

**IMPLEMENTASI METODE PARTICLE SWARM OPTIMIZATION-  
CERTAINTY FACTOR UNTUK PENGENALAN KONDISI IKAN  
LELE**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Sevtyan Eko Pambudi  
NIM: 125150202111005



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

IMPLEMENTASI METODE PARTICLE SWARM OPTIMIZATION-CERTAINTY FACTOR  
UNTUK PENGENALAN KONDISI IKAN LELE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Sevtyan Eko Pambudi  
NIM: 125150202111005

Skripsi ini telah diuji dan dinyatakan lulus pada  
2 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

  
Randy Cahya Wihandika, S.ST, M.Kom  
NIK: 201405 880206 1 001

  
Rekyan Regasari Mardi Putri, S.T, M.T  
NIK: 2011027704142001

Mengetahui  
Ketua Jurusan Teknik Informatika



  
Tri Astoro Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018



Sevtyan Eko Pambudi

NIM: 125150202111005



## KATA PENGANTAR

Dengan segala kerendahan hati dan diri memanjatkan syukur kehadiran Allah SWT atas limpahan rahmat dan hidayahNya sehingga dapat menyelesaikan penulisan skripsi dengan baik. Shalawat dan salam semoga senantiasa terlimpah curahan kepada Rasulullah Muhammad shallallahu`alaihiwasallam beserta keluarga dan sahabat-sahabat beliau.

Penulisan skripsi diajukan untuk memenuhi salah satu persyaratan dalam memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Brawijaya. Topik skripsi yang diajukan adalah **"Implementasi Metode Particle Swarm Optimization – Certainty Factor Untuk Pengenalan Kondisi Ikan Lele"**.

Kelancaran penulisan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis mengucapkan terimakasih dan rasa hormat kepada:

1. Bapak Randy Cahya Wihandika, S.ST, M.Kom selaku dosen pembimbing I yang telah memberikan arahan dan menyediakan waktu untuk berdiskusi dengan penulis selama proses penulisan skripsi.
2. Ibu Rekyan Regasari Mardi Putri, S.T, M.T selaku dosen pembimbing II yang telah memberikan banyak masukan dan motivasi kepada penulis selama proses penulisan skripsi.
3. Bapak Dedi selaku narasumber dari PT Indosco dalam wawancara tentang budidaya ikan lele beserta staff dan karyawan.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya
5. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Eko Sakti P., S.Kom, M.Kom selaku dosen penasehat akademik.
7. Dosen penguji
8. Kedua orang tua penulis, Bapak Tugiman dan Ibu Siti Khomariah. Terimakasih untuk doa, nasihat dan dukungan baik moril maupun materil.
9. Teman Penulis, Wahyu Argo Prabowo, Yanuardi Firmansyah, Dimas Angger Pribadi, Ullum Pratiwi, Ilham Rolis. Terimakasih untuk doa dan dukungan.
10. Fyma Ardita. Terimakasih atas semangat dan dukungan selama pengerjaan skripsi ini.
11. Seluruh *civitas academica* Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari kata sempurna, oleh karenanya segala kritik dan saran yang bersifat membangun sangat penulis harapkan demi perbaikan skripsi ini. Akhirnya, semoga karya

sederhana ini dapat memberikan manfaat bagi semua pihak sekaligus sebagai sarana pendukung pada pengembangan karya-karya terkait dimasa mendatang.

Malang, 2 Agustus 2018

Sevtyan Eko Pambudi

sevtyanekopambudi@gmail.com





## ABSTRAK

Sevtyan Eko Pambudi. 2018: Implementasi Metode *Particle Swarm Optimization – Certainty Factor* Untuk Pengenalan Kondisi Ikan Lele. Fakultas Ilmu Komputer Universitas Brawijaya. Pembimbing: Randy Cahya Wihandika, S.ST, M.Kom dan Rekyan Regasari Mardi Putri, S.T, M.T

Kegagalan dalam budidaya ikan lele sering kali disebabkan karena belum menemukan kombinasi komposisi terbaik saat ingin memulai budidaya seperti, jenis kolam, pemberian antiseptik, dll. Maka dari itu dibutuhkan suatu sistem yang bertujuan menemukan kombinasi parameter terbaik sekaligus memprediksi kondisi ikan sebelum diimplementasikan dalam dunia nyata. Salah satu metode yang dapat diterapkan untuk memprediksi kondisi ikan adalah *certainty factor*. Namun, kinerja *certainty factor* sangat tergantung kepada ahli yang berkaitan dengan permasalahan sehingga solusi yang dihasilkan rentan terjebak pada daerah optimum lokal. Salah satu pendekatan yang dapat digunakan untuk mengatasi permasalahan ini adalah dengan menerapkan algoritma optimasi yaitu *Particle Swarm Optimization (PSO)*. PSO mengeksplorasi ruang pencarian untuk menemukan nilai cf pakar berdasarkan nilai cost partikel. Nilai cost dirancang untuk meminimalkan jarak antara nilai random dengan nilai bobot sehingga semakin kecil mendekati 0 (nol) maka semakin besar peluang sebuah partikel terpilih sebagai solusi.

Penelitian ini menggunakan hybrid *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele. Kualitas *Certainty Factor* dievaluasi menggunakan data uji dari pakar dengan membandingkan keluaran sistem. Hasil percobaan menunjukkan bahwa algoritma hybrid *PSO-Certainty Factor* menghasilkan hasil prediksi lebih baik dibandingkan dengan algoritma *Certainty Factor* yaitu sebesar 90%.

Kata kunci: Prediksi, Optimasi, *Particle Swarm Optimization*, *Certainty Factor*, Ikan lele

## ABSTRACT

**Sevtyan Eko Pambudi. 2018: Implementation of the Particle Swarm Optimization - Certainty Factor Method to Recognize the Condition of Catfish. Faculty of Computer Science Universitas Brawijaya. Advisor: Randy Cahya Wihandika, S.ST, M.Kom and Rekyan Regasari Mardi Putri, S.T, M.T**

Failure in catfish farming is often caused by not finding the best composition combination when you want to start cultivation such as, types of ponds, antiseptic administration, etc. Therefore we need a system that aims to find the best combination of parameters while predicting the condition of the fish before it is implemented in the real world. One method that can be applied to predict fish conditions is certainty factor. However, the performance of certainty factor is highly dependent on experts related to the problem so that the resulting solution is vulnerable to being trapped in the local optimum area. One approach that can be used to overcome this problem is to apply optimization algorithms, namely Particle Swarm Optimization (PSO). PSO explores the search space to find the value of the expert cf based on the value of the particle cost. The value of the cost is designed to minimize the distance between random values and the weight value so that the smaller is close to 0 (zero) the greater the chance of a particle being selected as a solution.

This study uses hybrid Particle Swarm Optimization-Certainty Factor to identify the condition of catfish. The quality of Certainty Factor is evaluated using test data from experts by comparing system output. The experimental results show that the PSO-Certainty Factor hybrid algorithm produces better predictive results compared to the Certainty Factor algorithm which is 90%.

**Keywords:** Prediction, Optimization, Particle Swarm Optimization, Certainty Factor, Catfish.

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN .....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Budidaya Ikan Lele .....	7
2.3 Certainty Factor .....	8
2.4 Particle Swarm Optimization (PSO) .....	9
2.4.1 Komponen Particle Swarm Optimization.....	10
2.4.2 Langkah-langkah Algoritma PSO .....	11
2.5 Hybrid Particle Swarm Optimization-Certainty Factor .....	12
2.5.1 Penerapan Particle Swarm Optimization .....	13
2.5.2 Penerapan Certainty Factor .....	14
2.5.3 Pengujian Akurasi.....	14
BAB 3 METODOLOGI .....	15
3.1 Studi Literatur .....	16
3.2 Analisa Kebutuhan .....	16
3.3 Pengumpulan Data.....	17



3.4 Perncangan Sistem.....	17
3.5 Implementasi Sistem.....	17
3.6 Pengujian Sistem.....	18
3.7 Penarikan Kesimpulan .....	18
<b>BAB 4 PERANCANGAN.....</b>	<b>19</b>
4.1 Formulasi Permasalahan.....	19
4.2 Siklus Algoritma Particle Swarm Optimization dan CF .....	21
4.2.2 Inisialisasi Partikel .....	23
4.2.3 Perhitungan Nilai Cost.....	25
4.2.4 Update Kecepatan dan Posisi.....	26
4.2.5 Update Personal Best dan Global Best .....	28
4.2.6 Perhitungan CF Gejala.....	29
4.2.7 Perhitungan CF Kombinasi .....	30
4.3 Siklus Penyelesaian Masalah Menggunakan Metode PSO-CF.....	31
4.3.1 Inisialisasi Partikel .....	32
4.3.2 Perhitungan Nilai Cost Iterasi 0.....	32
4.3.3 Perhitungan Inisialisasi Personal Best dan Global Best .....	34
4.3.4 Perhitungan Update Kecepatan dan Posisi .....	34
4.3.5 Perhitungan Nilai Cost Iterasi 1.....	35
4.3.6 Perhitungan Update Personal Best dan Global Best.....	35
4.3.7 Perhitungan CF Gejala.....	36
4.3.8 Perhitungan CF Kombinasi .....	40
4.4 Perancangan Pengujian Algoritma.....	42
4.5 Perancangan Antarmuka .....	45
4.5.1 Perancangan Antarmuka User .....	46
4.5.2 Perancangan Antarmuka Budidaya.....	46
4.5.3 Perancangan Antarmuka Hasil Optimasi PSO .....	47
4.5.4 Perancangan Antarmuka Hasil Percobaan .....	48
4.5.5 Perancangan Antarmuka Kuis .....	48
4.5.6 Perancangan Antarmuka Admin Home .....	49
4.5.7 Perancangan Antarmuka Parameter Admin .....	50
4.5.8 Perancangan Antarmuka Perhitungan CF .....	50

4.6 Perancangan Database .....	51
4.6.1 Entity Relationship Diagram (ERD).....	51
BAB 5 IMPLEMENTASI .....	53
5.1 Implementasi Algoritma Hybrid PSO-CF .....	53
5.1.1 Implementasi Inisialisasi Partikel .....	53
5.1.2 Implementasi Menghitung Nilai Cost.....	54
5.1.3 Implementasi Personal Best.....	55
5.1.4 Implementasi Global Best .....	55
5.1.5 Implementasi Update Kecepatan dan Posisi Partikel .....	56
5.1.6 Implementasi Inisialisasi CF Pakar .....	57
5.1.7 Implementasi Perhitungan CF Gejala .....	57
5.1.8 Implementasi Perhitungan CF Kombinasi .....	58
5.2 Implementasi Antarmuka .....	59
5.2.1 Implementasi Antarmuka User .....	60
5.2.1 Implementasi Antarmuka Admin .....	62
5.3 Implementasi Database .....	63
BAB 6 PENGUJIAN .....	66
6.1 Pengujian Parameter PSO .....	66
6.1.1 Pengujian Bobot Inersia .....	66
6.1.2 Pengujian Kecepatan Akselerasi .....	69
6.1.3 Pengujian Jumlah Iterasi .....	71
6.1.4 Pengujian Ukuran Swarm.....	73
6.1.5 Pengujian Akurasi.....	75
BAB 7 PENUTUP .....	84
7.1 Kesimpulan.....	84
7.2 Saran .....	84
DAFTAR PUSTAKA.....	85
LAMPIRAN A DATA BUDIDAYA IKAN LELE.....	86

## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	5
Tabel 2.2 Nilai Interpretasi dari Pakar.....	6
Tabel 4.1 Fakta Budidaya Ikan Lele.....	19
Tabel 4.2 Indikasi Budidaya Ikan Lele.....	20
Tabel 4.3 Skema Inisialisasi Nilai Partikel.....	23
Tabel 4.4 Nilai Inisialisasi Partikel.....	32
Tabel 4.5 Nilai Partikel.....	33
Tabel 4.6 Nilai Bobot Pakar.....	33
Tabel 4.7 Hasil Nilai Cost.....	33
Tabel 4.8 Nilai Cost Terbaik.....	34
Tabel 4.9 Inisialisasi Personal Best.....	34
Tabel 4.10 Inisialisasi Global Best.....	34
Tabel 4.11 Perhitungan Update Kecepatan Partikel.....	34
Tabel 4.12 Konversi Kecepatan Partikel.....	35
Tabel 4.13 Perhitungan Update Posisi Partikel.....	35
Tabel 4.14 Perhitungan Nilai Cost Iterasi 1.....	35
Tabel 4.15 Perbandingan Cost Iterasi 0 dan Iterasi 1.....	36
Tabel 4.16 Update Personal Best Iterasi 1.....	36
Tabel 4.17 Update Global Best Iterasi 1.....	36
Tabel 4.18 Nilai CF Pakar Setelah Optimasi.....	37
Tabel 4.19 Nilai CF User.....	37
Tabel 4.20 Hasil Perhitungan CF Gejala.....	39
Tabel 4.21 Hasil Perhitungan CF Kombinasi.....	41
Tabel 4.22 Rancangan Pengujian Bobot Inersia.....	42
Tabel 4.23 Rancangan Pengujian Koefisien Akselerasi.....	43
Tabel 4.24 Rancangan Pengujian Jumlah Iterasi.....	44
Tabel 4.25 Rancangan Pengujian Ukuran Swarm.....	44
Tabel 6.1 Hasil Pengujian Bobot Inersia Maksimal 0,9.....	66
Tabel 6.2 Hasil Pengujian Bobot Inersia Maksimal 0,8.....	67
Tabel 6.3 Hasil Pengujian Bobot Inersia Maksimal 0,7.....	67
Tabel 6.4 Hasil Percobaan Koefisien Akselerasi = 1.....	69

Tabel 6.5 Hasil Percobaan Koefisien Akselerasi = 1,5.....	69
Tabel 6.6 Hasil Percobaan Koefisien Akselerasi = 2.....	70
Tabel 6.7 Hasil Pengujian Jumlah Iterasi.....	71
Tabel 6.8 Hasil Pengujian Ukuran Swarm.....	73
Tabel 6.9 Hasil Pengujian Akurasi Hybrid PSO-CF dengan Hasil Pakar.....	75
Tabel 6.10 Hasil Pengujian Akurasi CF dengan Hasil Pakar.....	79



## DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Metodologi Penelitian.....	16
Gambar 4.1 Diagram Alir Proses Algoritma PSO-CF.....	22
Gambar 4.2 Diagram Alir Proses Inisialisasi Partikel.....	24
Gambar 4.3 Diagram Alir Prsoses Perhitungan Cost.....	25
Gambar 4.4 Diagram Alir Proses Update Kecepatan dan Posisi.....	27
Gambar 4.5 Diagram Alir Proses Update Personal Best dan Global Best.....	29
Gambar 4.6 Diagram Alir Proses Perhitungan CF Gejala.....	30
Gambar 4.7 Diagram Alir Proses Perhitungan CF Kombinasi.....	31
Gambar 4.8 Perancangan Antarmuka Login.....	45
Gambar 4.9 Perancangan Antarmuka User Home.....	46
Gambar 4.10 Perancangan Antarmuka Budidaya.....	47
Gambar 4.11 Perancangan Antarmuka Hasil Optimasi PSO.....	47
Gambar 4.12 Perancangan Antarmuka Hasil Percobaan.....	48
Gambar 4.13 Perancangan Antarmuka Kuis.....	49
Gambar 4.14 Perancangan Antarmuka Admin Home.....	49
Gambar 4.15 Perancangan Antarmuka Parameter Admin.....	50
Gambar 4.16 Perancangan Antarmuka Perhitungan CF.....	50
Gambar 4.17 ERD Media Pembelajaran Budidaya Ikan Lele.....	51
Gambar 4.18 PDM Media Pembelajarn Budidaya Ikan Lele.....	52
Gambar 5.1 Kode Program Inisialisasi Partikel.....	53
Gambar 5.2 Kode Program Inisialisasi Menghitung Nilai Cost.....	54
Gambar 5.3 Kode Program Personal Best.....	55
Gambar 5.4 Kode Program Global Best.....	56
Gambar 5.5 Kode Program Update Kecepatan dan Posisi Partikel.....	57
Gambar 5.6 Kode Program Inisialisasi CF Pakar.....	57
Gambar 5.7 Kode Program Perhitungan CF Gejala.....	58
Gambar 5.8 Kode Program Perhitungan CF Kombinasi.....	58
Gambar 5.9 Implementasi Antarmuka Login.....	59
Gambar 5.10 Implementasi Antarmuka User Home.....	60
Gambar 5.11 Implementasi Antarmuka Budidaya.....	60
Gambar 5.12 Implementasi Antarmuka Hasil Optimasi PSO.....	61

Gambar 5.13 Implementasi Antarmuka Hasil Percobaan.....	61
Gambar 5.14 Implementasi Antarmuka Kuis.....	62
Gambar 5.15 Implementasi Antarmuka Admin Home.....	62
Gambar 5.16 Implementasi Antarmuka Parameter Admin.....	63
Gambar 5.17 Implementasi Antarmuka Certainty Factor.....	63
Gambar 5.18 Implementasi Database Media Pembelajaran Budidaya Lele.....	64
Gambar 6.1 Grafik Pengujian Bobot Inersia.....	64
Gambar 6.2 Grafik Pengujian Koefisien Akselerasi.....	70
Gambar 6.3 Grafik Pengujian Jumlah Iterasi.....	72
Gambar 6.4 Grafik Waktu Komputasi Berdasarkan Iterasi.....	72
Gambar 6.5 Grafik Pengujian Ukuran Swarm.....	74
Gambar 6.6 Grafik Pengujian Ukuran Swarm Terhadap Waktu.....	74





## DAFTAR LAMPIRAN

LAMPIRAN A DATA BUDIDAYA IKAN LELE.....	86
A.1 Data Fakta.....	86
A.2 Data Indikasi .....	87
A.3 Data Uji .....	87



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Budidaya ikan adalah kegiatan untuk memelihara, membesarkan dan/atau mengembangbiakkan ikan serta memanen hasilnya dalam lingkungan terkontrol (Depkeu, 2004). Budidaya ikan juga dapat memanfaatkan efisiensi lahan / tanah yang tidak cocok untuk pertanian atau perkebunan, jadi lahan tersebut dapat dimanfaatkan untuk budidaya ikan.

Salah satu ikan yang banyak dibudidayakan adalah ikan lele, bahkan ikan lele masuk dalam sepuluh komoditas utama budidaya. Tingkat produksi lele Nasional terus mengalami kenaikan dengan rata-rata mencapai 18,88% per tahun (Direktorat Jenderal Perikanan Budidaya, 2016). Selain kandungan gizi dan harganya yang murah tingginya produksi ikan lele di Indonesia didorong tingginya permintaan terhadap ikan lele, hal ini tentu menjadi peluang bisnis yang sangat menggiurkan.

Dalam membudidayakan ikan lele terdapat dua peran sekaligus, yang pertama orang membudidayakan lele hanya sebagai hiburan dan yang kedua membudidayakan lele dijadikan ladang bisnis. Bagi orang yang memelihara lele hanya sebagai hiburan tentu tidak terlalu diambil pusing jika ikan yang dihasilkan tidak maksimal karena yang dicari hanyalah kesenangan, tetapi bagi orang yang memilih lele sebagai bisnis penghasilan utama tentu menjadi masalah yang serius jika ikan yang dihasilkan tidak sesuai dengan yang ditargetkan.

Bagi seseorang yang baru ingin memulai budidaya ikan lele atau para pebisnis pemula yang beralih untuk membudidayakan ikan lele, sering kali mengalami kegagalan, hal ini disebabkan karena belum menemukan kombinasi komposisi terbaik dalam melakukan pembudidayaan ikan lele seperti pemilihan jenis kolam yang akan digunakan, pemberian antiseptik pada ikan dan minimnya pengetahuan mengenai budidaya ikan lele yang dimiliki. Untuk menggali pengetahuan dapat dilakukan dengan mengikuti pelatihan, membaca-baca buku referensi yang ada. Akan tetapi menggali pengetahuan melalui pelatihan, dan membaca buku referensi membutuhkan biaya dan waktu yang lebih lama sehingga dibutuhkan sebuah sistem yang dapat menyediakan fasilitas coba-coba seperti layaknya melakukan budidaya secara nyata didalamnya untuk mengenali kondisi ikan lele sebelum diimplementasikan dalam dunia nyata.

Dalam ilmu komputer salah satu metode yang dapat digunakan dalam memprediksi mengenali kondisi ikan lele yaitu metode *Certainty factor*. *Certainty factor* merupakan metode yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan, dengan menyatakan kepercayaan terhadap suatu kejadian atau hipotesis berdasarkan beberapa fakta yang terjadi dengan parameter coba-coba sebagai fakta (Yastita, Lulu, & Sari, 2013). Fakta dapat ditambahkan berupa perilaku, dan ciri-ciri umum pada ikan saat proses pembudidayaan. Contoh fakta perilaku ikan lemas, ciri – ciri umum pada ikan tampak bintik-bintik putih dsb. Dengan mengimplementasikan metode certainty

factor untuk mengenali kondisi ikan lele dapat memberikan pengetahuan dengan pengalaman coba-coba komposisi budidaya ikan lele.

Penelitian tentang *Certainty factor* telah banyak dilakukan dengan objek yang berbeda-beda. Penelitian oleh (Rohajawati & Supriyati, 2010), *Certainty factor* digunakan untuk mendiagnosis penyakit unggas (ayam). Pada penelitian tersebut *Certainty factor* mampu memberikan hasil yang akurat dari perhitungan bobot untuk kesimpulan diagnosis yang dihasilkan. Algoritme *Certainty factor* sangat sensitif terhadap konversi nilai kepercayaan suatu fakta terhadap hipotesis yang diperoleh dari seorang ahli atau pakar disebut dengan nilai cf pakar, dalam hal ini seorang ahli akan mengkonversikan pengetahuannya ke dalam bentuk angka. Nilai yang diberikan seorang pakar belum tentu merupakan nilai paling akurat karena penilaian dari pakar satu dengan pakar yang lain dalam satu keahlian bisa berbeda, dan tentunya akan mempengaruhi hasil akhir dari algoritme. Dalam ilmu komputer ada metode yang dapat menutupi kelemahan dari *Certainty factor* dengan cara pembangkitan nilai secara acak dengan nilai evaluasi sebagai acuannya yaitu metode *Particle Swarm Optimization*. Dengan optimasi yang dilakukan oleh PSO menjadikan nilai cf pakar yang diperoleh dari ahli lebih akurat dan hasil dari algoritme *Certainty factor* lebih maksimal.

Metode *Particle Swarm Optimization* banyak diterapkan pada beberapa masalah yang kompleks. Penelitian oleh Permana & Hashim (2010) menerapkan PSO untuk menyelesaikan optimasi pada keanggotaan *fuzzy* dalam hal ini mencari nilai keanggotaan *fuzzy* yang paling akurat. Hasil dari penelitian tersebut mendekati optimum dengan waktu yang cukup cepat. Hal tersebut dikarenakan PSO mempunyai kelebihan yaitu mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan jika dibandingkan dengan algoritme matematika dan teknik optimasi heuristik lainnya (Tuegeh, 2009).

Kajian dari beberapa penelitian diatas menunjukkan bahwa algoritme *Particle Swarm Optimization* dapat mengoptimasi nilai cf pakar pada algoritme *Certainty Factor*. Sehingga panggabungan kedua metode dapat meningkatkan kinerja dari algoritme *Certainty Factor* dengan menjadikan nilai cf pakar lebih optimal, maka hasil akhir perhitungan juga lebih optimal.

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, penulis mengajukan penelitian yang berjudul "Implementasi Metode *Particle Swarm Optimization-Certainty Factor* Untuk Mengenali Kondisi Ikan Lele". Dalam penelitian ini diharapkan dapat memberikan pengetahuan yang lebih untuk pemula mengenai budidaya ikan lele.

## 1.2 Rumusan masalah

1. Bagaimana mengimplementasikan *Hybrid Particle Swarm Optimization* dan *Certainty Factor* untuk mengenali kondisi ikan lele?
2. Bagaimana tingkat akurasi penggunaan *Hybrid Particle Swarm Optimization* dan *Certainty Factor* dibandingkan dengan *Certainty Factor* untuk mengenali kondisi ikan lele?

### 1.3 Tujuan

Tujuan:

1. Merancang dan mengimplementasikan perangkat lunak menggunakan *Hybrid Particle Swarm Optimization – Certainty Factor* untuk mengenali kondisi ikan lele.
2. Mengukur tingkat akurasi *Hybrid Particle Swarm Optimization* dan *Certainty Factor* dibandingkan dengan *Certainty Factor* untuk mengenali kondisi ikan lele.

### 1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat secara teoritis, sekurang-kurangnya dapat berguna sebagai sumbangan pemikiran bagi dunia pendidikan dan dapat dijadikan sebagai bahan pertimbangan atau dikembangkan lebih lanjut, serta referensi terhadap penelitian yang sejenis. Bagi pembudidaya lele khususnya pemula hasil penelitian ini dapat memberikan manfaat sebagai ilmu pengetahuan tambahan mengenai ikan lele.

### 1.5 Batasan masalah

Batasan masalah pada penelitian ini adalah

1. Sistem dirancang menggunakan bahasa pemrograman PHP dan database MySQL.
2. Data yang digunakan sebanyak 40 fakta dalam budidaya salah satunya pemberian antiseptik dan lima jenis indikasi yaitu terserang bakteri, terserang parasit, ikan sehat, keracunan dan mati.

### 1.6 Sistematika pembahasan

Sistematika penulisan laporan penelitian ini disusun menjadi beberapa bab sebagai berikut:

#### BAB I PENDAHULUAN

Pada bab ini berisi pendahuluan yang menjelaskan latar belakang, perumusan masalah, tujuan, batasan masalah, dan sistematika pembahasan.

#### BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi landasan teori sebagai parameter rujukan untuk dilaksanakannya penelitian ini. Adapun landasan teori tersebut adalah hasil penelitian terkait dan *Particle Swarm Optimization-Certainty Factor*.

### BAB III METODOLOGI

Pada bab ini menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi dari metode *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele.

### BAB IV PERANCANGAN

Pada bab ini membahas perancangan metode *Particle Swarm Optimization-Certainty Farctor* untuk mengenali kondisi ikan lele.

### BAB V IMPLEMENTASI

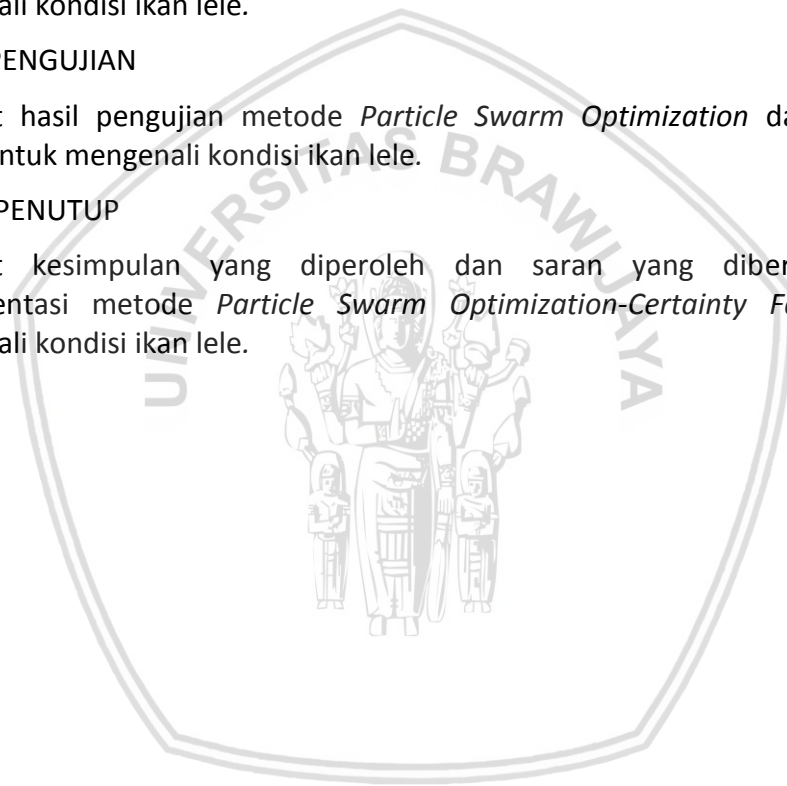
Bab ini membahas proses penerapan dan tahap-tahap yang dilakukan dalam penerapan metode *Particle Swarm Optimization* dan *Certainty Factor* untuk mengenali kondisi ikan lele.

### BAB VI PENGUJIAN

Memuat hasil pengujian metode *Particle Swarm Optimization* dan *Certainty Factor* untuk mengenali kondisi ikan lele.

### BAB VII PENUTUP

Memuat kesimpulan yang diperoleh dan saran yang diberikan untuk implementasi metode *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai kajian pustaka dan dasar-dasar teori yang akan digunakan dalam penulisan tugas akhir tentang implementasi metode *Hybrid Particle Swarm Optimization – Certainty Factor* untuk mengenali kondisi ikan lele. Kajian pustaka memberikan informasi terkait penelitian-penelitian yang sudah pernah dilakukan dan tentunya memiliki hubungan dengan sistem yang akan dibangun. Beberapa teori yang dibutuhkan adalah teori yang berkaitan dengan ikan lele sebagai objek penelitian, dan metode *Particle Swarm Optimization, Certainty Factor* sebagai algoritme yang akan diimplementasikan dalam penelitian.

### 2.1 Kajian Pustaka

Pada penulisan skripsi ini kajian pustaka berisi mengenai penelitian-penelitian sebelumnya yang berkaitan dengan objek atau metode sesuai dengan judul yang diajukan implementasi metode *Particle Swarm Optimization – Certainty Factor* untuk mengenali kondisi ikan lele.

Salah satu penelitian mengenai penggunaan metode *Certainty Factor* pada diagnosis penyakit tanaman jagung. Dalam penelitian ini metode *Certainty Factor* mampu mendiagnosa penyakit jagung berdasarkan gejala-gejala yang diinputkan oleh user. Tingkat akurasi hasil penelitian ini mencapai 86,67% (Majid, 2015).

Penelitian selanjutnya yang dilakukan mengenai optimasi komposisi pakan ikan lele. Penelitian ini menerapkan metode *Multi Objective Optimization with Invasive Weed Optimization-Subtractive Clustering* (IWO-SC). Hasil dari penelitian tersebut membuktikan metode IWO-SC dapat menghasilkan komposisi pakan lele yang optimal dengan peningkatan efisiensi biaya pakan sebesar Rp 922,00/Kg (Muliantara, 2012).

Penelitian selanjutnya mengenai penggunaan metode *Particle Swarm Optimization* untuk menghasilkan generasi pada fungsi keanggotaan *fuzzy*. Dalam penelitian ini metode *Fuzzy Particle Swarm Optimization (FPSO)* berhasil menunjukkan kompetensinya dalam fungsi keanggotaan *fuzzy*. Hal tersebut dilakukan dengan merepresentasikan nilai keanggotaan *fuzzy* sebagai partikel. Dalam setiap iterasi pada metode *PSO*, partikel tersebut akan berubah untuk mencari nilai optimal. Berdasarkan penelitian tersebut, dengan metode *FPSO* fungsi keanggotaan *fuzzy* yang dihasilkan dapat optimal meningkatkan kinerja serta lebih akurat (Permana & Hashim, 2010).

Penelitian selanjutnya mengenai penggunaan hybrid metode *K-Means Clustering* dan *PSO*. *PSO* digunakan untuk mengoptimasi titik pusat tiap cluster yang dibuat berdasarkan data. Objek yang diteliti yaitu data nasabah kredit dengan 13 parameter yang telah ditentukan. Dalam implementasinya, juga menggunakan metode *Random Injection* untuk mengatasi terjadinya konvergensi dini. Hasil akhir, metode hybrid *PSO-K-means* memberikan solusi yang lebih baik dibandingkan *K-means* (Anggodo, et al., 2017).



**Tabel 2.1 Kajian Pustaka**

No	Judul	Objek	Metode	Hasil
1	Implementasi Metode <i>Certainty Factor</i> Pada Diagnosis Penyakit Tanaman Jagung (Majid, 2015)	Tanaman Jagung	<i>Certainty Factor</i>	Metode <i>Certainty Factor</i> dapat diimplementasikan untuk mendiagnosa penyakit jagung, dengan tingkat akurasi kesesuaian mencapai 86,67%.
2.	Penentuan Komposisi Bahan Pakan Ikan Lele yang Optimal dengan Menggunakan Metode <i>Invasive Weed Optimization(IWO)-Subtractive Clustering</i>	Pakan Ikan Lele	<i>Invasive Weed Optimization (IWO)-Subtractive Clustering</i>	Parameter yang digunakan yaitu jumlah dimensi : 4, populasi awal : 20, populasi maksimum : 100, modulation indeks : 1 , acceptance ration : 0.5, rejection ratio : 0.15, iterasi maksimum : 5000, solusi counter : 10.  Komposisi pakan ikan lele optimal dan tingkat efisiensi biaya pakan sebesar Rp 922,00 /Kg.
3.	<i>Fuzzy Membership Function Generation using Particle Swarm Optimizaton</i> (Permana & Hashim, 2010)	Truck Backer Upper	<i>Fuzzy Particle Swarm Optimizaton</i>	Fungsi keanggotaan <i>Fuzzy</i> optimal sehingga dapat meningkatkan kinerja serta hasil yang didapatkan lebih akurat.
4.	Hybrid K-means dan <i>Particle Swarm Optimization</i> untuk <i>Clustering</i> Nasabah Kredit	Nasabah Kredit	<i>PSO, K-means Clustering, dan Random Injection</i>	Penggunaan hybrid <i>PSO</i> K-Means menjadikan titik pusat cluster lebih optimal dan dapat melakukan clustering data dengan benar sebesar 80%.
5	Usulan Penulis:	Ikan Lele	<i>Particle Swarm</i>	Proses algoritma yang digunakan dalam

	Implementasi metode PSO-Certainty Factor untuk Mengetahui Kondisi Ikan Lele.		Optimizaton (PSO) dan Certainty Factor	sistem mampu menghasilkan prediksi dari kombinasi fakta ikan lele dengan Certainty Factor yang sebelumnya nilai cf pakar sudah di perbaiki oleh algortima <i>Particle Swarm Optimization</i> .
--	--	--	--	--

## 2.2 Budidaya Ikan Lele

Dalam budidaya ikan lele terdapat tiga kegiatan utama yang dilakukan yaitu pembenihan, pendederan, dan pembesaran, ketiga kegiatan tersebut saling berhubungan (Agriminakultura, 2008). Secara umum pembenihan merupakan kegiatan dalam berbudidaya untuk menghasilkan benih dengan cara mengawinkan induk jantan dan betina. Proses pembesaran benih sampai berukuran tertentu dari hasil pembenihan dinamakan pendederan, dan selanjutnya dilakukan proses pembesaran sampai ikan lele memiliki ukuran siap untuk dikonsumsi (Agriminakultura, 2008).

Lele merupakan salah satu komoditas unggulan. Pengembangan usahanya dapat dilakukan mulai dari benih sampai ukuran konsumsi. Ukuran siap jual yang umumnya berlaku di kalangan peternak lele adalah 1-3 cm untuk benih, 3-5 cm untuk pendederan 1, 5-8 cm untuk pendederan 2, dan untuk konsumsi berat minimal ikan lele 200 gram per ekor (Khairuman dan Amri 2008).

Setiap segmen usaha ini sangat menguntungkan. Selain untuk konsumsi lokal, pasar lele telah mulai di ekspor dan permintaannya cukup besar. Bahkan pada tahun 2011 Indonesia menempati posisi teratas yang mendominasi produksi lele dunia dengan memberikan *share* sekitar 75,6 % terhadap total produksi ikan lele dunia (Direktorat Jenderal Perikanan Budidaya, 2013). Tingkat produksi lele konsumsi secara Nasional meningkat setiap tahunnya. Pada tahun 2015 produksi lele sebesar 719.619 ton (Direktorat Jenderal Perikanan Budidaya, 2016). Pada tahun 2016 produksi lele sebesar 873.716 ton, atau meningkat rata-rata 18,88 % pertahun (Direktorat Jenderal Perikanan Budidaya, 2016). Tingkat kebutuhan lele juga meningkat pesat. Pada tahun 2015 dibutuhkan 1.058.400 ton, pada tahun 2016 dibutuhkan 1.217.100 ton, sedangkan pada akhir tahun 2019 diperkirakan akan dibutuhkan 1.770.600 ton atau meningkat 13,75 % per tahun (Direktorat Jenderal Perikanan Budidaya, 2016).

Jenis lele yang banyak dibudidayakan di Indonesia dan dijumpai di pasaran saat ini adalah ikan lele dumbo (*Clarias Gariepinus*) (Khairuman dan Amri 2008). Dalam kegiatan budidaya secara intensif, ikan lele didorong untuk tumbuh secara maksimum hingga mencapai ukuran optimal. Lele dumbo merupakan komoditas yang dapat dipelihara dengan padat tebar tinggi dalam lahan terbatas dan hemat

air. Untuk kolam ukuran 15 m<sup>2</sup> lele dumbo dapat ditebar sebanyak 5.250 ekor benih. Selama 2,5 bulan dapat diproduksi lele sebanyak 450 kg dengan nilai fcr (Fed Caonversion Ratio) satu (Direktorat Jenderal Perikanan Budidaya, 2013).

### 2.3 Certainty Factor

Faktor kepastian (certainty factor) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN. Certainty factor (CF) merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. Formula dasar Certainty factor didefinisikan sebagai berikut :

$$CF(H, E) = MB(H, E) - MD(H, E) \quad (2.1)$$

Keterangan :

CF(H, E): certainty factor dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara -1 sampai 1. Nilai -1 menunjukkan ketidakpercayaan mutlak, sedangkan nilai 1 menunjukkan kepercayaan mutlak.

MB(H,E) : ukuran kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

MD(H,E) : ukuran ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E (antara 0 dan 1).

Formula dasar digunakan apabila belum ada nilai CF untuk setiap gejala yang menyebabkan indikasi. Kombinasi *certainty factor* yang digunakan untuk kaidah dengan gejala atau fakta tunggal adalah (Turban, 2005):

$$Cf_{gejala} = Cf_{user} \times Cf_{pakar} \quad (2.2)$$

Keterangan :

$Cf_{gejala}$  : Nilai kepercayaan untuk setiap fakta atau gejala yang mempengaruhi suatu indikasi.

$Cf_{user}$  : Nilai kepercayaan yang diberikan oleh pengguna terhadap fakta atau gejala yang dialami.

$Cf_{pakar}$  : Nilai kepercayaan yang diberikan oleh seorang pakar untuk setiap fakta atau gejala yang mempengaruhi indikasi.

Apabila gejala atau fakta lebih dari satu, maka perhitungan menggunakan *certainty factor combine* yang ditunjukkan persamaan 2.3 (Turban, 2005).

$$Cf_{combine} = Cf_{old} + Cf_{gejala} \times (1 - Cf_{old}) \quad (2.3)$$

Keterangan :

$Cf_{combine}$  : Nilai kepercayaan hasil kombinasi dari beberapa fakta atau gejala.

$Cf_{old}$  : Nilai kepercayaan dari fakta pertama atau fakta sebelumnya.

$Cf_{gejala}$  : Nilai kepercayaan dari fakta kedua atau fakta selanjutnya.

Untuk hasil akhir dari algoritma *certainty factor* yang menunjukkan persentase terhadap suatu indikasi menggunakan persamaan 2.4 sebagai berikut.

$$Cf_{persentase} = Cf_{combine} \times 100 \quad (2.4)$$

Keterangan :

$Cf_{persentase}$  : Persentase nilai *certainty factor* terhadap suatu indikasi.

$Cf_{combine}$  : Hasil akhir nilai *certainty factor combine* dari beberapa  $Cf_{gejala}$ .

Nilai CF (Rule) didapat dari interpretasi “term” dari pakar, yang diubah menjadi nilai CF tertentu sesuai dengan tabel berikut:

**Tabel 2.2 Nilai Interpretasi dari pakar**

No	<i>Uncertain Term</i>	CF
1	<i>Definitely not</i> (pasti tidak)	-1.0
2	<i>Almost certainly not</i> (hampir pasti tidak)	-0.8
3	<i>Probably not</i> (kemungkinan besar tidak)	-0.6
4	<i>Maybe not</i> (mungkin tidak)	-0.4
5	<i>Unknown</i> (tidak tahu)	-0.2 to 0.2
6	<i>Maybe</i> (mungkin)	0.4
7	<i>Probably</i> (kemungkinan besar)	0.6
8	<i>Almost certainly</i> (hampir pasti)	0.8
9	<i>Definitely</i> (pasti)	1

## 2.4 Particle Swarm Optimization (PSO)

Algoritma *Particle Swarm Optimization* (PSO) pertama kali dikenalkan oleh Kennedy dan Eberhart pada tahun 1995. Algoritma PSO merupakan sebuah teknik optimasi yang berbasis populasi untuk mencari solusi optimal menggunakan populasi dari partikel itu sendiri. PSO didasarkan pada ide bahwa setiap kerumunan partikel merupakan sebuah solusi dari ruang solusi. Dalam PSO populasi disebut dengan *swarm*, dan individu disebut dengan *particle* (Permana & Hashim, 2010). Pada awal algoritma dijalankan, dia akan membangkitkan sejumlah “kandidat solusi” sebanyak  $n$ . Dalam PSO kandidat solusi disebut partikel/burung, sejumlah  $n$  disebut *swarm*/populasi. Pada prosesnya solusi ini diperbarui dengan adanya iterasi atau dalam PSO partikel terbang mengelilingi ruang pencarian untuk menemukan solusi terbaik. Proses menentukan solusi terbaik yaitu dengan fungsi evaluasi yang terdiri dari *fitness* dan *cost*. Menggunakan fungsi *fitness* ketika tujuannya memaksimalkan nilai

evaluasi sehingga semakin besar nilai evaluasi semakin bagus dan fungsi *cost* dengan meminimalkan nilai evaluasi untuk menemukan solusi terbaik. Pada penelitian ini menggunakan fungsi *cost* untuk meminimalkan jarak solusi dengan nilai bobot. Partikel/burung akan selalu terbang disepanjang ruang pencarian sampai dengan iterasi maksimal. Dapat diasumsikan setiap burung berhenti pada suatu titik, dia sekaligus menghitung nilai evaluasi (*fitness/cost*) kemudian terbang lagi sampai iterasi maksimal. Setiap partikel/burung akan mendapatkan posisi terbaik pada setiap iterasinya atau yang disebut sebagai personal best, sementara posisi terbaik dari semua partikel disebut global best. Semuanya akan dievaluasi berdasarkan nilai *fitness* atau *cost*, sehingga partikel akan terus terbang memeriksa setiap titik (calon solusi) berdasarkan nilai *fitness* atau *cost* yang dihasilkan sampai dengan iterasi maksimal. Kemudian saat terbang, partikel punya kecepatan untuk menuju titik tertentu. Beberapa penelitian ada yang mendefinisikan pembatasan kecepatan dan ada yang tidak, selain itu besar kecepatan juga berbeda-beda tergantung permasalahan yang dihadapi. Untuk menentukan kecepatan salah satu cara menggunakan percobaan-percobaan dan pengalaman.

Kesederhanaan algoritma dan performansinya yang baik, menjadikan PSO telah menarik banyak perhatian di kalangan para peneliti dan telah diaplikasikan dalam berbagai persoalan optimisasi sistem. Berikut pada subbab 2.5.1 akan dijelaskan komponen-komponen dari algoritma PSO.

#### 2.4.1 Komponen Particle Swarm Optimization

Komponen algoritma PSO adalah sebagai berikut (Tueguh, 2009):

- a. Swarm  
Swarm merupakan jumlah partikel dalam populasi pada suatu algoritma. Ukuran swarm tergantung pada seberapa kompleks masalah yang dihadapi.
- b. Partikel  
Partikel merupakan individu dalam suatu swarm yang merepresentasikan calon solusi penyelesaian masalah. Setiap partikel memiliki kecepatan dan posisi yang ditentukan oleh representasi solusi pada saat itu.
- c. Personal best (pBest)  
Personal best merupakan posisi terbaik yang pernah dicapai partikel dengan membandingkan nilai *cost* pada posisi partikel sekarang dan sebelumnya. Personal best dipersiapkan untuk mendapatkan solusi terbaik.
- d. Global best (gBest)  
Global best merupakan posisi terbaik partikel yang diperoleh dengan membandingkan nilai *cost* terbaik dari keseluruhan partikel dalam swarm.
- e. Kecepatan (Velocity)  
Velocity merupakan vector yang menentukan arah perpindahan posisi sebuah partikel. Perubahan velocity dilakukan setiap iterasi dengan tujuan memperbaiki posisi partikel.



## 2.4.2 Langkah-langkah Algoritma PSO

Algoritma PSO terdiri dari beberapa tahap, yaitu pembangkitan partikel, pembangkitan posisi serta kecepatan partikel, *update velocity*, *update position*. Partikel berubah posisinya dari suatu perpindahan (iterasi) ke posisi lainnya berdasarkan pada *update velocity*.

Langkah pertama adalah pembangkitan partikel, dimana pada tahap ini dibangkitkan nilai random sesuai dengan permasalahan masing-masing. Nilai random ini yang nantinya akan menjadi kandidat solusi dari algoritma ini seiring pembaruan iterasi.

Langkah kedua adalah posisi  $x_k^i$  dan kecepatan  $v_k^i$  dari kumpulan partikel dibangkitkan secara *random (rand)* menggunakan batas atas ( $X_{max}$ ) dan batas bawah ( $X_{min}$ ) seperti yang ditunjukkan pada persamaan 2.5 dan 2.6.

$$x_k^i = X_{min} + rand (X_{max} - X_{min}) \quad (2.5)$$

$$v_k^i = X_{min} + rand (X_{max} - X_{min}) \quad (2.6)$$

langkah selanjutnya adalah *update velocity* (kecepatan) untuk semua partikel pada waktu  $k+1$  menggunakan nilai *cost* posisi partikel saat ini pada saat waktu ke  $k$ .

Dalam penelitian ini, nilai *cost* digunakan untuk meminimalkan jarak solusi dengan nilai *cf* pakar. Nilai *cf* pakar merupakan nilai paling optimum yang di dapat dari hasil wawancara terhadap pakar. Jarak yang paling optimum adalah jumlah partikel yang mendekati 0(nol) namun tidak menutup kemungkinan hasil yang didapat bernilai 0(nol) seiring dengan update kecepatan dan posisi. Solusi yang di maksud adalah nilai partikel atau kandidat solusi yang di bangkitkan secara acak 0(nol) sampai 1. Nilai *cost* dicari menggunakan persamaan 2.7 sebagai berikut:

$$\text{Nilai Cost} = (\sum_{j=1}^n \text{Nilai bobot} \times \text{Nilai tiap partikel}) - \text{Nilai bobot} \quad (2.7)$$

Dari nilai *cost* dapat ditentukan partikel yang memiliki nilai global terbaik (*global best*) pada *swarm* saat ini, dan juga posisi terbaik dari tiap partikel pada semua waktu yang sekarang dan sebelumnya. Perumusan *update velocity* juga menggunakan beberapa parameter *random*. Perumusan *update velocity* dapat dilihat pada persamaan 2.8 sebagai berikut (Valle, 2008).

$$v_{k+1}^i = \omega * v_k^i + c_1 * rand * p_{best}^i - x_k^i + c_2 * rand * g_{best} - x_k^i \quad (2.8)$$

dimana :

$v_k^i$  = kecepatan patikel i pada iterasi k

$\omega$  = *inertia* (fungsi pemberat)

$c_1$  = *self confidence*

$c_2$  = *swarm confidence*

*rand* = nilai acak antara 0 dan 1



$x_k^i$  = posisi partikel i pada iterasi k

$p_{best}^i$  = posisi terbaik dari partikel i

$g_{best}$  = nilai  $p_{best}$  terbaik dari *swarm*.

Pada setiap iterasi, nilai fungsi pemberat (*inertia*) di-update melalui persamaan 2.9 (Bansal, 2011).

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} x \text{ iter} \quad (2.9)$$

Dimana :

$\omega_{max}$  = nilai pemberat (*inertia*) awal

$\omega_{min}$  = nilai pemberat (*inertia*) akhir

$iter_{max}$  = jumlah iterasi maksimum

$iter$  = jumlah iterasi terakhir.

Langkah terakhir adalah *update* posisi tiap partikel. Dengan adanya perubahan kecepatan, maka posisi partikel juga akan berubah pada tiap iterasi yang dapat dicari melalui persamaan 2.10 (Bansal, 2011).

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (2.10)$$

Dimana :

$x_{k+1}^i$  = posisi partikel saat ini

$x_k^i$  = posisi partikel sebelumnya

$v_{k+1}^i$  = kecepatan partikel saat ini.

Tiga langkah yang telah dipaparkan tersebut akan diulang sampai kriteria kekonvergenan terpenuhi. Kriteria kekonvergenan sangat penting dalam menghindari penambahan nilai evaluasi setelah solusi optimum didapatkan. Namun kriteria kekonvergenan tidak selalu mutlak diperlukan. Penetapan jumlah iterasi maksimal juga dapat digunakan sebagai *stopping condition* dari sebuah algoritma (Bansal, 2011). Dalam penelitian ini *stopping condition* yang digunakan adalah jumlah iterasi maksimal sehingga algoritma tidak akan berhenti sebelum jumlah iterasi selesai.

## 2.5 Hybrid Particle Swarm Optimization – Certainty Factor

*Hybrid Particle Swarm Optimization – Certainty Factor* merupakan algoritma hasil penggabungan antara *Particle Swarm Optimization* dan *Certainty Factor*. Algoritma ini dilakukan secara sekuensial artinya algoritma PSO dijalankan terlebih dahulu sampai pada kondisi tertentu berhenti kemudian dilanjutkan dengan algoritma *Certainty Factor*.

*Certainty Factor* merupakan salah satu metode pada ilmu komputer yang dapat digunakan memprediksi suatu hasil yang belum diketahui, dengan menyatakan kepercayaan terhadap suatu hipotesis berdasarkan beberapa fakta

yang ada kaitanya. Namun algoritma ini memiliki kelemahan yaitu sangat bergantung pada nilai kepercayaan yang diberikan oleh seorang pakar. Sedangkan antara pakar satu dengan pakar yang lain memiliki kapasitas yang berbeda, sehingga solusi yang dihasilkan rentan terjebak pada daerah optimum lokal.

Pada penelitian ini, PSO digunakan untuk menginisialisasi nilai kepercayaan yang diberikan oleh pakar (Cf Pakar). Hal tersebut bertujuan bahwa nilai Cf pakar yang digunakan pada *certainty factor* nantinya merupakan nilai yang paling mendekati optimum. Sehingga, jika nilai Cf pakar tepat maka dapat menyempurnakan proses algoritma *certainty factor* dengan harapan dapat memberikan hasil mendekati akurasi dari ahli atau bahkan lebih.

### 2.5.1 Penerapan Particle Swarm Optimization

PSO digunakan untuk mengoptimasi nilai kepercayaan yang diberikan oleh pakar (Cf pakar) algoritma *certainty factor*. Optimasi bertujuan untuk mendapatkan nilai Cf pakar yang paling optimum berdasarkan nilai cost setiap partikel. Langkah-langkah algoritma PSO untuk diterapkan pada penelitian ini sebagai berikut:

1. Inisialisasi Partikel  
Panjang partikel adalah  $f \times i$  ( $f$  = jumlah fakta dan  $i$  = jumlah indikasi) sehingga data dengan 40 fakta jika gabungkan ke banyaknya indikasi maka memiliki panjang partikel 200 sel dengan 40 sel terhadap indikasi terserang bakteri, 40 sel terhadap indikasi terserang parasit, 40 sel terhadap ikan sehat, 40 sel terhadap keracunan, dan 40 sel terhadap ikan mati. Panjang partikel tergantung pada masukan pengguna.
2. Menghitung nilai *cost* menggunakan Persamaan 2.7  
Perhitungan nilai *cost* digunakan untuk evaluasi sehingga dengan keadaan tertentu posisi nilai cf pakar berpindah menuju optimum sampai iterasi maksimal.
3. Update kecepatan dan posisi partikel menggunakan Persamaan 2.8 dan Persamaan 2.10.  
*Update* kecepatan pada penelitian ini digunakan untuk memperbarui posisi partikel setiap iterasinya berdasarkan kecepatan. Kecepatan dikonversikan bertujuan perpindahan posisi partikel tidak melewati ruang pencarian. Dalam penelitian ini batas ruang pencarian adalah nilai 1 berdasarkan nilai cf pakar maksimal.
4. Update Personal Best dengan ketentuan sebagai berikut :

$$(t+1) \begin{cases} p_i(t) & f(x_i(t+1)) \leq f(x_i(t)) \\ x_i(t+1) & f(x_i(t+1)) \leq f(x_i(t)) \end{cases}$$

Keterangan :

$(t+1)$  : *Personal best* pada iterasi  $t+1$

$x_i(t+1)$  : Partikel pada iterasi  $t+1$

$f(xi(t + 1))$  : Nilai *cost* partikel pada iterasi  $t+1$

5. Update Global Best.

Proses *update global best* adalah mencari nilai *personal best* yang memiliki nilai *cost* terbaik.

6. Ulangi langkah ke 2 sampai kriteria kondisi tertentu terpenuhi (maksimal iterasi).

### 2.5.2 Penerapan Certainty Factor

Penerapan *Certainty factor* digunakan untuk memprediksi dari hasil simulasi dan coba-coba. Simulasi dan coba-coba meliputi 40 fakta budidaya ikan lele dan 5 indikasi yang didapat dari hasil wawancara dengan ahli. Dari masing-masing fakta terdapat tingkat kepercayaan fakta terhadap suatu indikasi yang disebut nilai  $cf$  pakar. Nilai  $cf$  pakar didapat dari pembangkitan nilai secara acak perhitungan PSO dengan nilai evaluasi tertentu. Algoritma *certainty factor* dijalankan setelah algoritma PSO selesai. Langkah-langkah algoritma *certainty factor* yang diterapkan pada penelitian ini sebagai berikut:

1. Inisialisasi nilai  $Cf_{pakar}$  didapat dari pembangkitan nilai pada proses PSO.
2. Menghitung nilai  $Cf_{gejala}$  dari nilai  $Cf_{pakar}$  hasil optimasi dan  $Cf_{user}$  yang dimasukkan pengguna sebelumnya dengan Persamaan 2.2.
3. Melakukan perhitungan kombinasi dari nilai  $Cf_{gejala}$  untuk semua indikasi menggunakan Persamaan 2.3.
4. Menghitung persentase dari  $Cf_{combine}$  terhadap semua indikasi menggunakan Persamaan 2.4.
5. Indikasi dengan tingkat kepercayaan tertinggi menjadi hasil akhir.

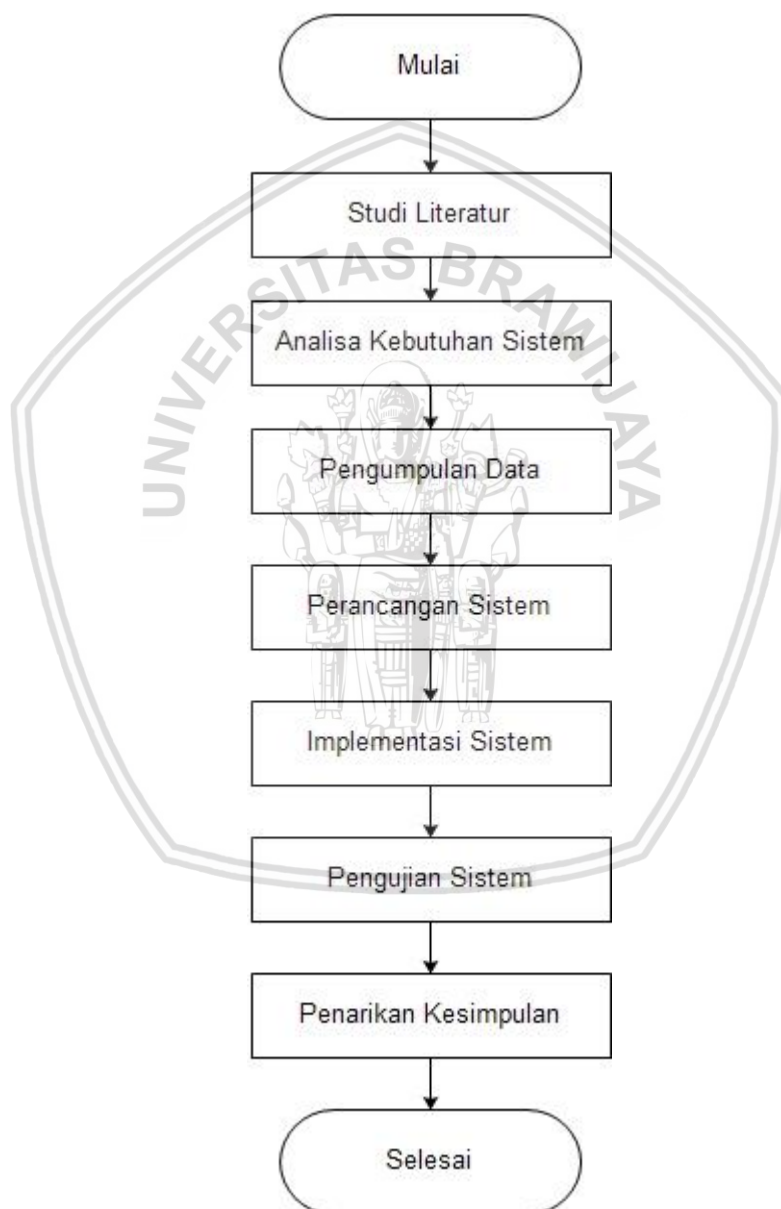
### 2.5.3 Pengujian Akurasi

Pengujian akurasi dilakukan dengan membandingkan hasil keluaran sistem dengan hasil dari pakar. Pengujian ini bertujuan untuk mengetahui peningkatan solusi sebelum hybrid PSO-Certainty factor dan sesudah hybrid PSO-Certainty factor. Dalam pengujian akurasi disediakan beberapa kasus uji yang akan di jawab oleh pakar berdasarkan pengetahuan dan dari sistem. Hasil akurasi dari hybrid PSO-Certainty factor di bandingkan dengan hasil akurasi sebelum hybrid yaitu menggunakan metode *Certainty Factor*. Dalam mengetahui nilai akurasi dilakukan mencocokkan hasil keluaran sistem dengan hasil dari pakar untuk mendapatkan nilai akurasi sistem. Persamaan untuk mencari nilai akurasi dapat dilihat pada Persamaan 2.11.

$$\text{Nilai akurasi} = \frac{\text{Jumlah data akurat}}{\text{jumlah seluruh data uji}} \times 100 \% \quad (2.11)$$

### BAB 3 METODOLOGI

Pada bab ini dijelaskan langkah-langkah yang dilakukan untuk mengerjakan penelitian tentang implementasi metode *Particle Swarm Optimization* dan *Certainty Factor* untuk mengenali kondisi ikan lele. Dengan adanya metodologi penelitian ini diharapkan dapat memberikan petunjuk dalam merumuskan masalah penelitian. Alur pengerjaan penelitian ini secara umum dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Diagram Alir Metodologi Penelitian**

### 3.1 Studi Literatur

Studi literatur mempelajari mengenai dasar teori yang digunakan untuk menunjang penulisan serta pengerjaan tugas akhir. Sumber yang digunakan sebagai studi literatur diperoleh dari buku, jurnal, e-book dan penelitian sebelumnya yang berkaitan tentang topik tugas akhir ini. Teori yang berkaitan dengan implementasi metode *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele, di antaranya:

1. Metode *Particle Swarm Optimization*
2. Metode *Certainty Factor*
3. Sistem Pakar
4. Budidaya ikan lele
5. Pemrograman dengan menggunakan Bahasa PHP
6. Pengujian Sistem

### 3.2 Analisa Kebutuhan

Analisa kebutuhan merupakan sebuah fase yang berfungsi untuk merumuskan kebutuhan – kebutuhan yang diperlukan dalam implementasi metode PSO-Certainty Factor untuk mengenali kondisi ikan lele.

Secara keseluruhan, kebutuhan yang digunakan dalam implementasi metode PSO-Certainty Factor untuk mengenali kondisi ikan lele ini meliputi:

1. Spesifikasi kebutuhan hardware, meliputi:
  - a. Laptop ASUS A46CB.
  - b. RAM 8 GB.
  - c. Harddisk 500GB.
  - d. Monitor 14”.
2. Spesifikasi kebutuhan software, meliputi:
  - a. Microsoft Windows 7 64bit sebagai sistem operasi.
  - b. Microsoft Office 2013.
  - c. Sublime Text 3 sebagai editor untuk membuat sistem.
  - d. PHP merupakan bahasa pemrograman untuk pembuatan sistem.
  - e. XAMPP sebagai server localhost.
  - f. MySQL sebagai Database Management System (DBMS).
  - g. Edraw Max 7.9 sebagai aplikasi untuk membuat diagram

### 3.3 Pengumpulan Data

Dalam penelitian ini, data yang digunakan dalam penelitian adalah data primer berupa data 40 fakta salah satunya adalah Kolam terpal dan 5 indikasi yaitu terserang parasit, terserang bakteri, ikan sehat, keracunan, dan ikan mati. Semua fakta-fakta yang ada di dapat dari parameter jenis kolam, persiapan air, seleksi benih, pemberian pakan, kualitas air, dan kontrol penyakit melalui wawancara dan observasi.

Lokasi penelitian di PT Indosco Sidoarjo. Indosco mempunyai budidaya ikan lele untuk mengimplementasikan obat ikan yang di produksi di perusahaan utama. Di lokasi penelitian lebih dari 12 bak dengan diameter 3 meter untuk tempat budidaya ikan lele.

### 3.4 Perancangan Sistem

Perancangan sistem dibuat berdasarkan hasil analisa kebutuhan dan pengumpulan data yang telah dilakukan. Perancangan ini dilakukan untuk mempermudah proses implementasi, pengujian serta analisis. Langkah-langkah dilakukan dalam perancangan sistem adalah sebagai berikut:

1. Penyelesaian permasalahan  
Penyelesaian permasalahan meliputi proses langkah-langkah algoritma *PSO – Certainty factor* yang digunakan untuk pemecahan masalah yang ada, dan direpresentasikan dalam bentuk diagram dan perhitungan secara sederhana.
2. Perancangan antarmuka pengguna  
Perancangan antarmuka pengguna bertujuan agar antarmuka sistem yang dibangun dapat memudahkan penggunaanya.
3. Perancangan pengujian  
Perancangan pengujian sistem yang dilakukan meliputi proses uji coba *PSO* yaitu banyak partikel, jumlah iterasi, *Wmax*, dan *Wmin*, serta pengujian akurasi pada hasil yang dihasilkan sistem.

### 3.5 Implementasi Sistem

Implementasi sistem adalah tahap membangun sistem yang sesuai dengan perancangan yang telah dibuat dan menerapkan hal yang telah didapatkan pada proses studi literatur. Sistem di optimasi menggunakan algoritma *Particle Swarm Optimization-Certainty Factor* untuk memberikan hasil yang baik. Sistem akan dibangun menggunakan pemograman PHP (*hypertext Preprocessor*) sebagai platform pengembangannya. Tahap implementasi yang dilakukan meliputi implementasi antarmuka, basis data, dan algoritma. Keluaran yang dihasilkan dari implementasi sistem ini adalah hasil dari kombinasi fakta yang belum pernah diketahui berdasarkan kepercayaan *certainty factor* dengan nilai cf pakar yang telah dioptimasi.



### 3.6 Pengujian Sistem

Pada tahap ini dilakukan pengujian sistem yang telah dibuat untuk mengetahui bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dan kebutuhan. Beberapa uji coba yang akan dilakukan untuk mengevaluasi program ini adalah:

1. Uji coba untuk menentukan jumlah iterasi PSO yang optimal.
2. Uji coba untuk menentukan jumlah partikel PSO yang optimal.
3. Uji coba untuk menentukan koefisien akselerasi yang optimal.
4. Uji coba untuk menentukan bobot inersia maksimum dan inersia minimum yang optimal.
5. Pengujian akurasi yang didapat dari perbandingan hasil keluaran dari sistem dengan pakar.

### 3.7 Penarikan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan dari perancangan, implementasi, dan pengujian dari implementasi metode PSO-Certainty Factor untuk mengenali kondisi ikan lele telah dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang diterapkan. Penarikan kesimpulan dilakukan untuk menjawab rumusan permasalahan yang sudah ditetapkan sebelumnya. Tahap terakhir dari penulisan adalah saran yang berkenaan dengan hasil yang telah dicapai yang berguna untuk memperbaiki kesalahan dalam pengembangan lebih lanjut kedepannya.

## BAB 4 PERANCANGAN

Pada bab ini menjelaskan formulasi permasalahan, siklus algoritma *Particle Swarm Optimization (PSO)-Certainty factor*, siklus penyelesaian implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele, perhitungan manual, serta perancangan antarmuka.

### 4.1 Formulasi Permasalahan

Pada sub bab formulasi permasalahan menjelaskan tentang permasalahan yang akan diselesaikan. Permasalahan yang diselesaikan adalah bagaimana memprediksi hasil kombinasi fakta yang dialami selama budidaya ikan lele agar dapat menentukan indikasi yang terjadi pada ikan lele menggunakan *PSO-Certainty Factor*. Fakta-fakta dan indikasi budidaya ikan lele menggunakan *PSO-Certainty Factor* dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

**Tabel 4.1 Fakta Budidaya Ikan Lele**

Kode Fakta	Nama Fakta
F1	Kolam Fiber
F2	Kolam Cor Beton
F3	Kolam Bata
F4	Kolam Terpal
F5	Kolam Tanah
F6	Obat Sterilisasi air Sedikit (0,1 – 0,3 mm)
F7	Obat Sterilisasi air Banyak (0,3 mm keatas)
F8	Penebaran Fermentasi diatas 500 ml
F9	Ukuran Benih dibawah 2 cm
F10	Ukuran Benih Datas 2 cm
F11	Keseragaman bibit dibawah 80%
F12	Keseragaman bibit diatas 80%
F13	Air tampak keruh
F14	Suhu dibawah 30 Derajat
F15	Suhu diatas 30 Derajat
F16	Pembuangan kotoran 2-3 kali / minggu
F17	Pembuangan kotoran lebih dari 4 kali / minggu
F18	Pemberian anti septik 1-2 kali/ minggu
F19	Pemberian anti septik lebih dari 2 kali/minggu

F20	Pemberian prebiotic 0-30 ppm/minggu
F21	Terdapat amoniak di air
F22	PH air asam
F23	Ukuran kepala terlalu besar
F24	Tulang sirip ikan runcing
F25	Pergerakan ikan lambat
F26	Terdapat bahan organik di kolam
F27	Ikan mengambang di atas
F28	Air tercemar H <sub>2</sub> S
F29	Kanibalisme tinggi
F30	Ukuran bibit ikan ideal
F31	PH air berubah ubah
F32	Bau semen
F33	Ikan sering dipermukaan bawah
F34	Pertumbuhan ikan lambat
F35	Ikan lebih aktif
F36	Pertumbuhan ikan lebih cepat
F37	Pemberian prebiotik lebih dari 30ppm/minggu
F38	Muncul lender di tubuh
F39	Warna ikan pucat
F40	Benjolan bewarna merah di sirip

**Tabel 4.2 Indikasi Budidaya Ikan Lele**

Kode Indikasi	Nama Indikasi
P1	Terserang Bakteri
P2	Terserang Parasite
P3	Ikan Sehat
P4	Keracunan
P5	Ikan Mati

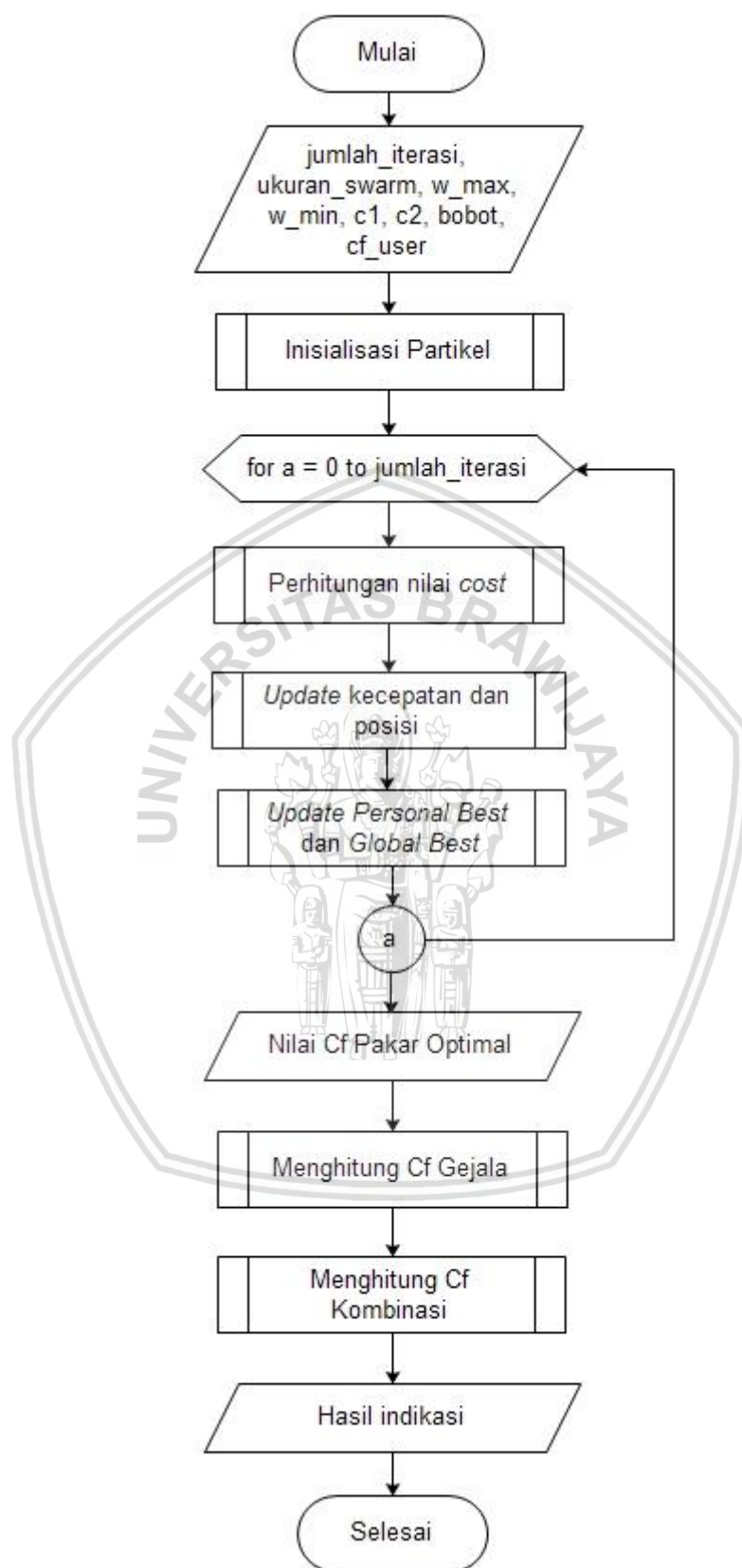
Dari tabel 4.1 dan 4.2 dapat di ambil contoh, seorang pembudidaya pemula ingin menggali pengetahuan tentang budidaya ikan lele. Karena keterbatasan

waktu dan biaya, pembudidaya memutuskan untuk langsung mencoba coba berdasarkan komposisi yang dia pikirkan seperti jenis kolam yang dipilih atau mengkombinasikan fakta secara coba-coba seperti ukuran memberi antiseptik, jenis kolam yang digunakan dst. Dengan melakukan seperti itu, dapat disimpulkan pembudidaya cenderung ingin mengetahui kondisi ikan lele yang terjadi dengan komposisi yang sudah diterapkan secara coba coba dari semua kombinasi fakta. Dengan harapan pembudidaya mendapatkan pengetahuan baru sehingga bisa meminimalkan kesalahan dalam mengambil langkah maupun kombinasi fakta yang akan digunakan secara nyata.

Solusi permasalahan dari paragraph sebelumnya adalah dengan sistem yang telah dirancang sebelumnya. Sistem dapat bekerja dengan masukan antara lain fakta yang dialami selama budidaya ikan lele dan parameter perhitungan optimasi meliputi: jumlah iterasi, ukuran swarm, bobot inersia maksimal ( $\omega_{max}$ ), bobot inersia minimal ( $\omega_{min}$ ), koefisien akselerasi 1 ( $C_1$ ), dan koefisien akselerasi 2 ( $C_2$ ). Selanjutnya masuk pada proses penyelesaian terdapat dua tahap. Tahap pertama adalah menentukan nilai cf pakar menggunakan algoritma *Particle Swarm Optimization*. Nilai cf pakar dihasilkan berdasarkan nilai *cost* terbaik dengan nilai terkecil mendekati 0 (nol). Keluaran algoritma *PSO* adalah nilai cf pakar. Tahap kedua adalah proses menentukan indikasi menggunakan algoritma *Certainty Factor*. Inisialisasi nilai cf pakar diperoleh dari hasil optimasi menggunakan algoritma *PSO*. Selanjutnya proses kombinasi nilai cf pakar dan cf user yang sebelumnya diinputkan oleh pengguna dengan hasil kombinasi kepercayaan tertinggi sebagai kesimpulan sistem.

#### **4.2 Siklus Algoritma Particle Swarm Optimization (PSO) dan Certainty Factor**

Siklus algoritma *PSO* dan *Certainty Factor* merupakan urutan penyelesaian masalah menggunakan algoritma *Particle Swamn Optimization-Certainty Factor* secara sekuensial. Optimasi menggunakan algoritma *PSO* bertujuan untuk memperoleh nilai cf pakar yang optimum berdasarkan nilai *cost* yang dihasilkan oleh partikel sebagai representasi solusi. Nilai cf pakar yang dihasilkan dapat memberikan nilai kepercayaan terhadap proses algoritma *Certainty factor*. Diagram alir proses diagnosa menggunakan *PSO* dan *Certainty Factor* ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Proses Algoritma PSO-Certainty Factor

Langkah-langkah algoritma PSO-Certainty Factor berdasarkan Gambar 4.1 adalah:

1. Sistem menerima masukan berupa parameter *PSO* dan *Certainty factor* meliputi: jumlah iterasi, ukuran swarm, bobot inersia maksimal ( $\omega_{max}$ ), bobot inersia minimal ( $\omega_{min}$ ), koefisien akselerasi 1 ( $C_1$ ), koefisien akselerasi 2 ( $C_2$ ), bobot, dan cf user.
2. Melakukan proses inialisasi partikel.
3. Melakukan perhitungan *cost*.
4. Melakukan *update* kecepatan dan posisi.
5. Melakukan *update personal best* dan *global best*.
6. Untuk iterasi 1 sampai jumlah iterasi, lakukan langkah 3 sampai langkah 5.
7. Keluaran dari proses optimasi adalah nilai  $Cf_{pakar}$  optimal.
8. Melakukan perhitungan cf tiap gejala ( $Cf_{gejala}$ ).
9. Melakukan perhitungan cf kombinasi ( $Cf_{combine}$ ).
10. Keluaran sistem adalah indikasi dengan  $Cf_{combine}$  terbesar dari algortima *Certainty factor* sebagai keputusan akhir.

#### 4.2.2 Inialisasi Partikel

Inialisasi partikel pada penelitian ini menggunakan partikel yang memiliki panjang sesuai dengan jumlah fakta yang diinputkan. Tiap sel merupakan representasi dari bobot masing-masing  $Cf_{pakar}$  yang dibangkitkan secara acak dengan *range* 0 sampai 1. Contoh skema inialisasi partikel dengan 3 fakta dan langkah-langkah inialisasi partikel ditunjukkan pada Tabel 4.3 dan Gambar 4.2.

**Tabel 4.3 Skema Inialisasi Partikel**

P	A			B			C			D			E		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
2	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
3	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
4	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
5	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

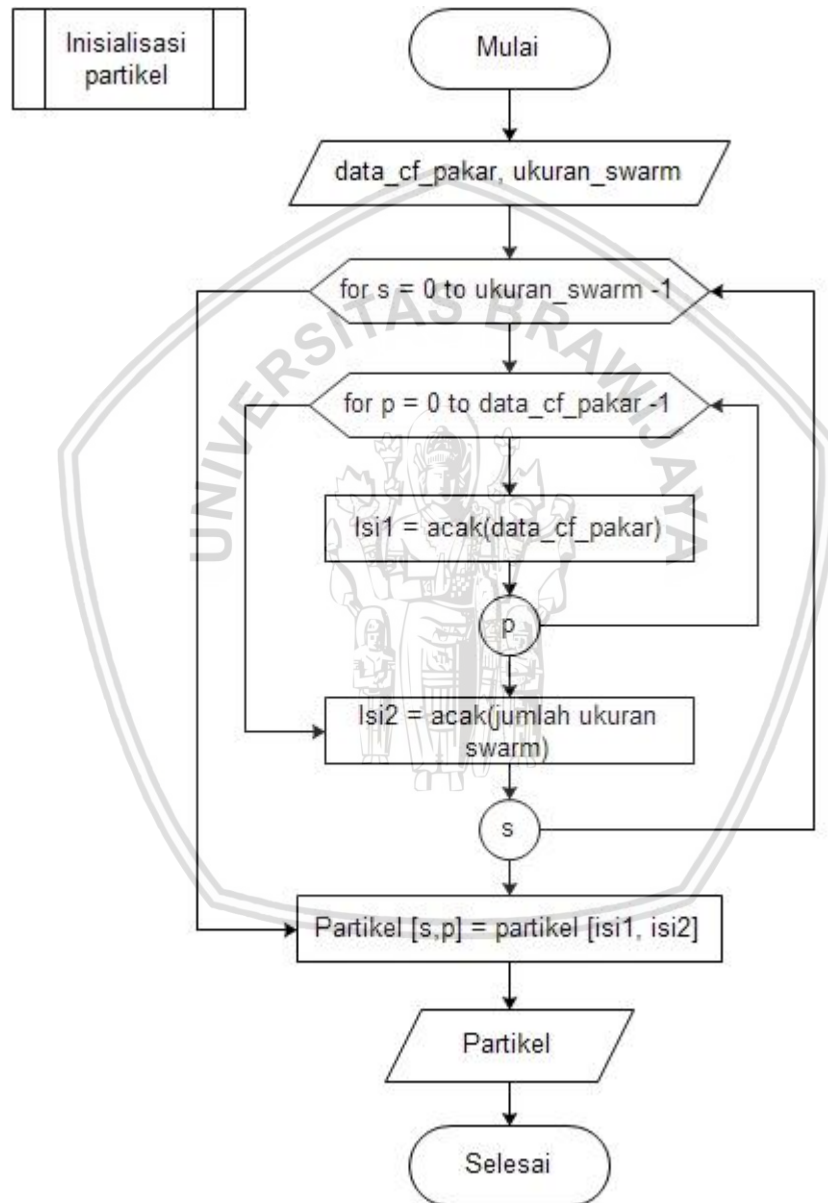
Keterangan :

- A : Indikasi 1  
 B : Indikasi 2  
 C : Indikasi 3  
 D : Indikasi 4  
 E : Indikasi 5  
 P : Partikel ke-  
 F1 : Fakta 1



F2 : Fakta 2  
F3 : Fakta 3  
N : Nilai acak 0 sampai 1.

Diagram alir proses inisialisasi partikel pada algoritma *PSO* ditunjukan pada Gambar 4.2.



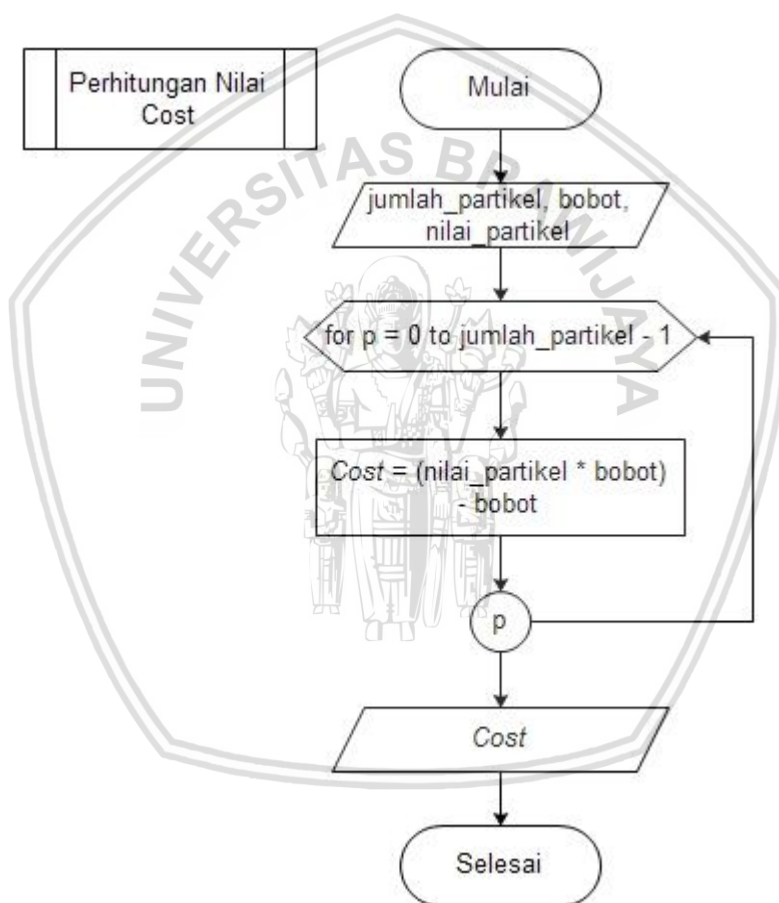
**Gambar 4.2 Diagram Alir Proses Inisialisasi Partikel**

Langkah-langkah inisialisasi partikel pada algoritma *PSO* berdasarkan Gambar 4.2 adalah sebagai berikut:

1. Sistem menerima masukan berupa data  $Cf_{pakar}$  dan ukuran swarm.
2. Untuk tiap partikel sampai jumlah swarm, lakukan langkah 3.
3. Untuk tiap nilai  $Cf_{pakar}$  dari partikel diperoleh secara acak untuk membangkitkan bobot nilai pakar tersebut.
4. Keluaran sistem adalah partikel PSO sebanyak jumlah swarm.

#### 4.2.3 Perhitungan Nilai Cost

Setiap partikel akan menghasilkan nilai *cost* untuk menentukan seberapa optimum solusi yang didapatkan. Pada penelitian ini nilai *cost* digunakan untuk mengetahui jarak antara *cost* yang dihasilkan dengan nilai bobot dari pakar sehingga nilai *cost* yang baik yaitu jarak terkecil atau mendekati 0 (nol). Diagram alir perhitungan nilai *cost* ditunjukkan pada Gambar 4.3.



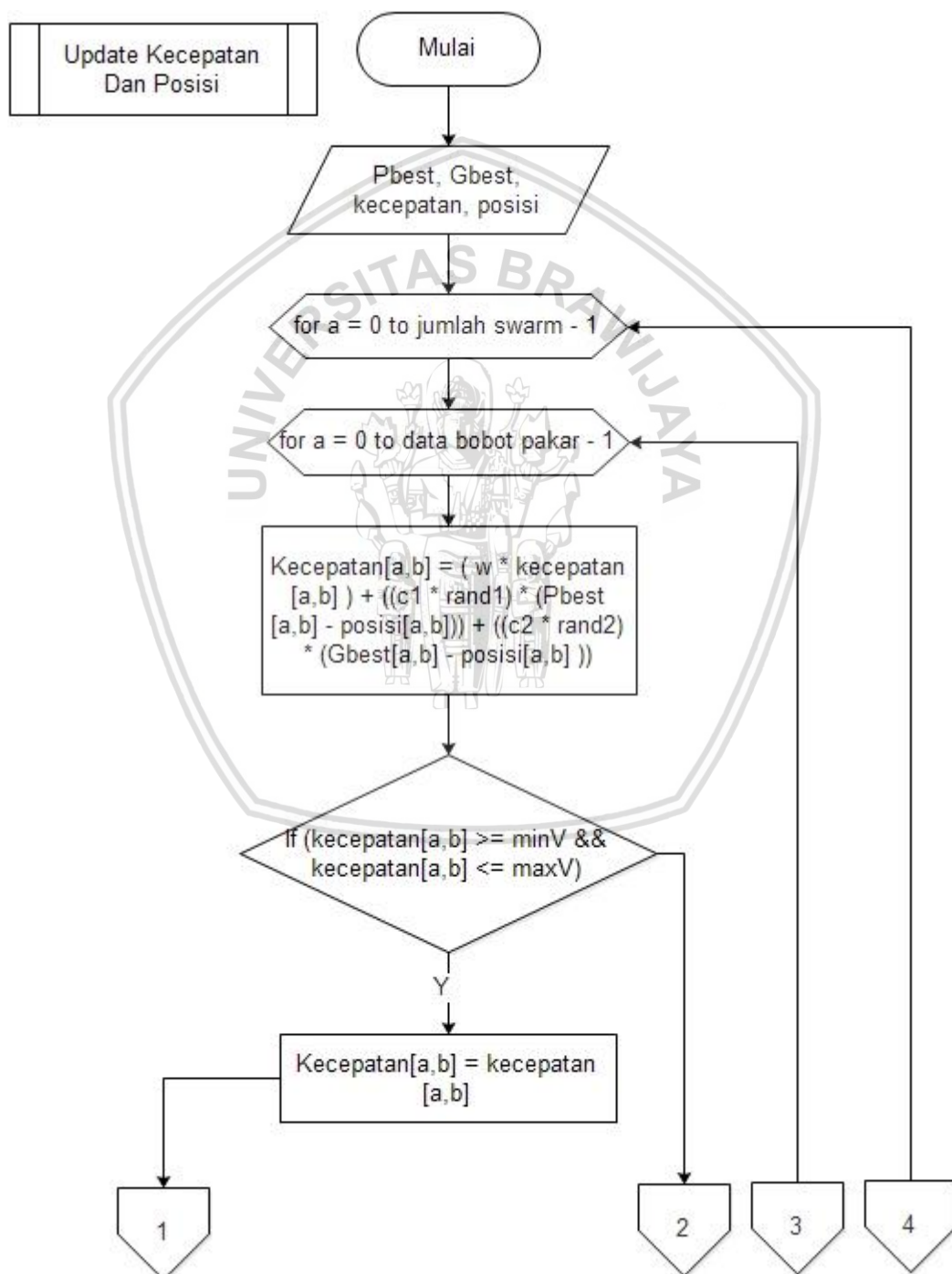
**Gambar 4.3 Diagram Alir Proses Perhitungan Cost.**

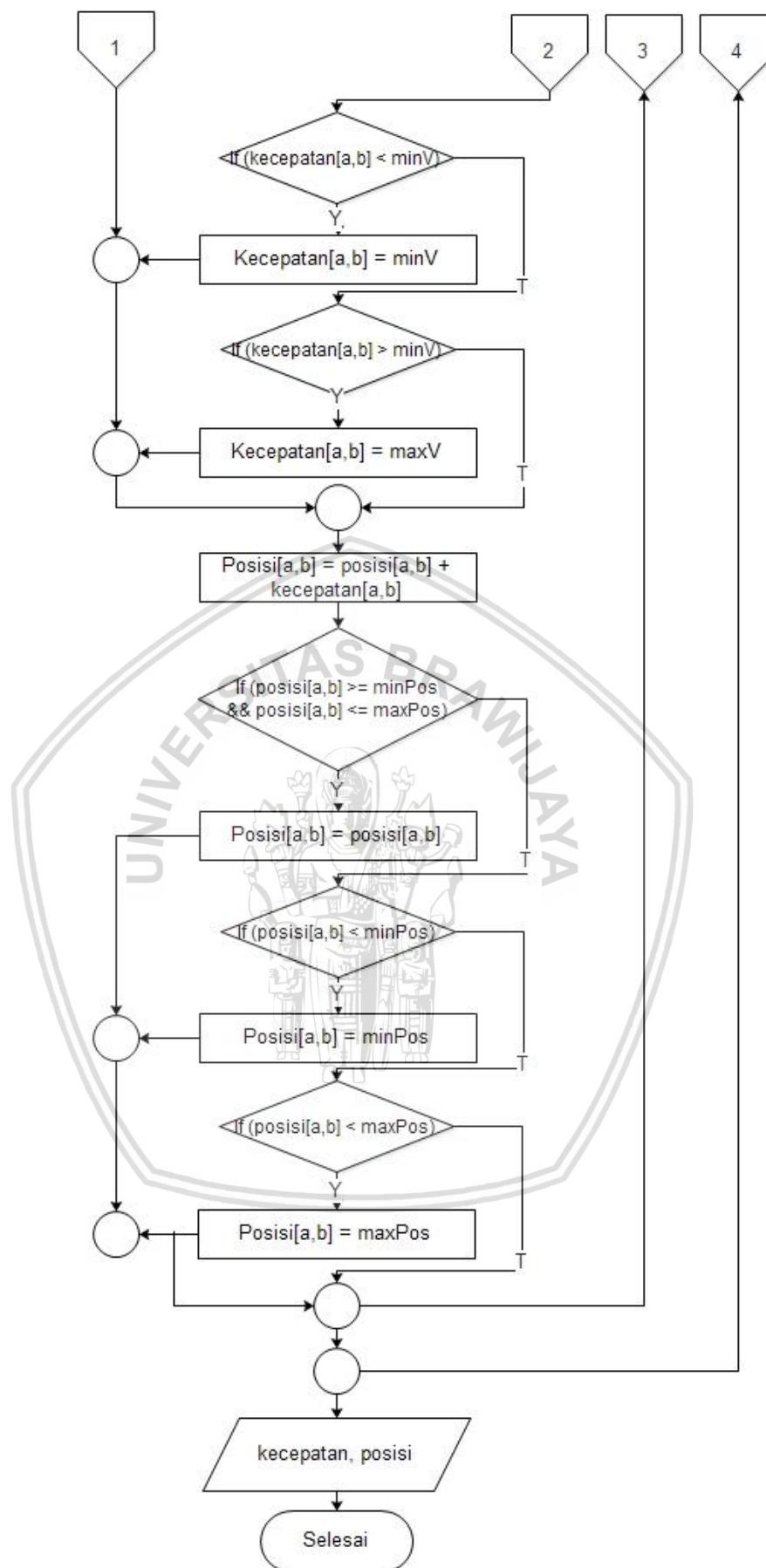
Langkah-langkah perhitungan nilai *cost* pada algoritma PSO berdasarkan Gambar 4.3 adalah sebagai berikut:

1. Sistem menerima masukan berupa jumlah partikel, bobot dan nilai partikel.
2. Menghitung nilai *cost* menggunakan Persamaan 2.6.
3. Ulangi langkah 2 sampai jumlah partikel – 1.
4. Hasil dari proses ini adalah nilai *cost* dari tiap partikel.

#### 4.2.4 Update Kecepatan dan Posisi

Pada tahap inisialisasi, nilai kecepatan adalah 0 (nol) dan nilai posisi sama dengan nilai inisialisasi partikel yang dibangkitkan secara acak. Namun pada iterasi berikutnya, nilai kecepatan dan posisi dihitung berdasarkan persamaan 2.7 dan 2.9. Tujuan update kecepatan dan posisi adalah dengan setiap iterasi partikel diharapkan semakin menemukan posisi yang optimum. Diagram alir update kecepatan dan posisi ditunjukkan pada Gambar 4.4.





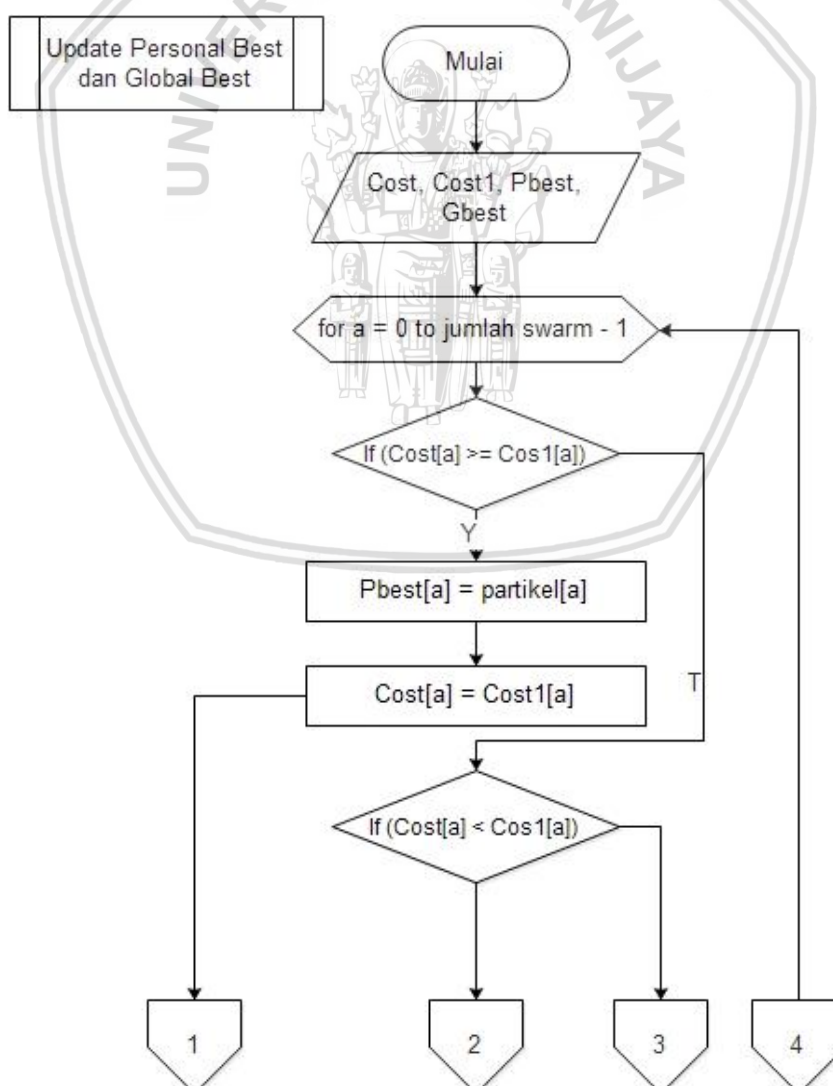
Gambar 4.4 Diagram Alir Proses *Update* Kecepatan dan Posisi.

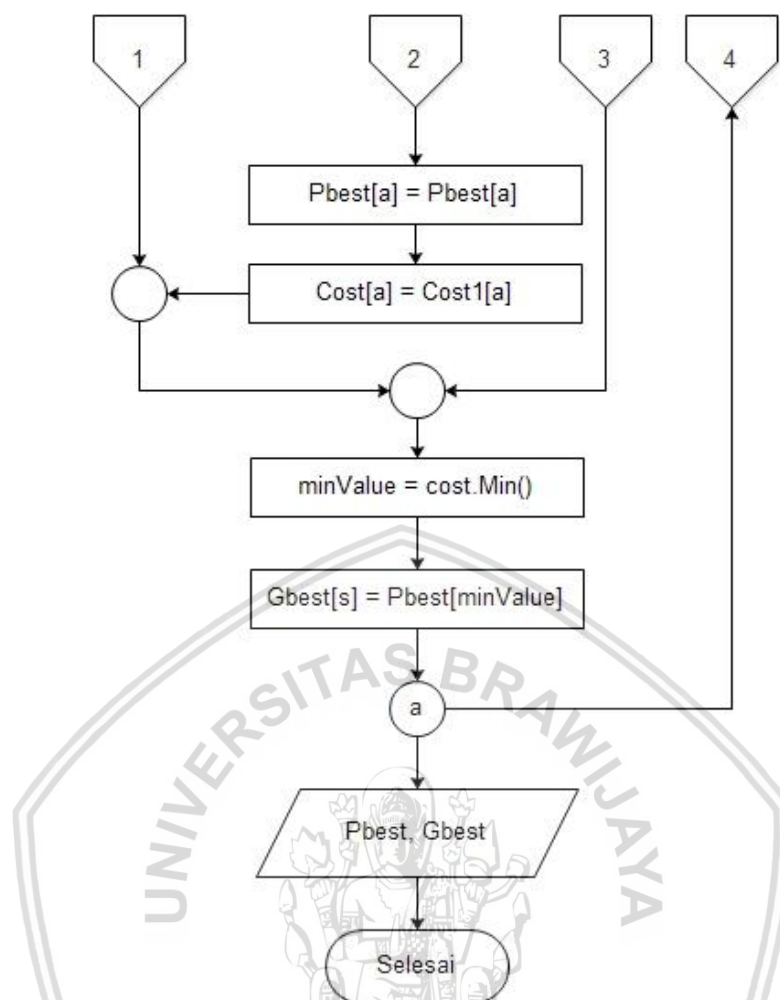
Langkah-langkah *update* kecepatan dan posisi pada algoritma *PSO* berdasarkan Gambar 4.4 adalah sebagai berikut:

1. Sistem menerima masukan berupa personal best, global best, kecepatan, dan posisi partikel.
2. Melakukan update kecepatan menggunakan Persamaan 2.7.
3. Melakukan update posisi menggunakan Persamaan 2.9.
4. Hasil dari proses ini adalah kecepatan dan posisi partikel yang telah diperbaharui.

#### 4.2.5 Update Personal Best dan Global Best

Pada tahap inisialisasi, nilai personal best dan global best ditentukan berdasarkan nilai inisialisasi partikel sedangkan global best ditentukan berdasarkan nilai personal best dengan nilai cost terkecil mendekati 0 (nol). Pada iterasi berikutnya, nilai personal best diperoleh dengan membandingkan nilai fitness personal best sekarang dengan nilai iterasi sebelumnya. Sedangkan global best diperoleh berdasarkan nilai personal best dengan cost terkecil. Diagram alir update personal best dan global best ditunjukkan pada Gambar 4.5.





**Gambar 4.5 Diagram Alir Proses *Update Personal Best* dan *Global Best*.**

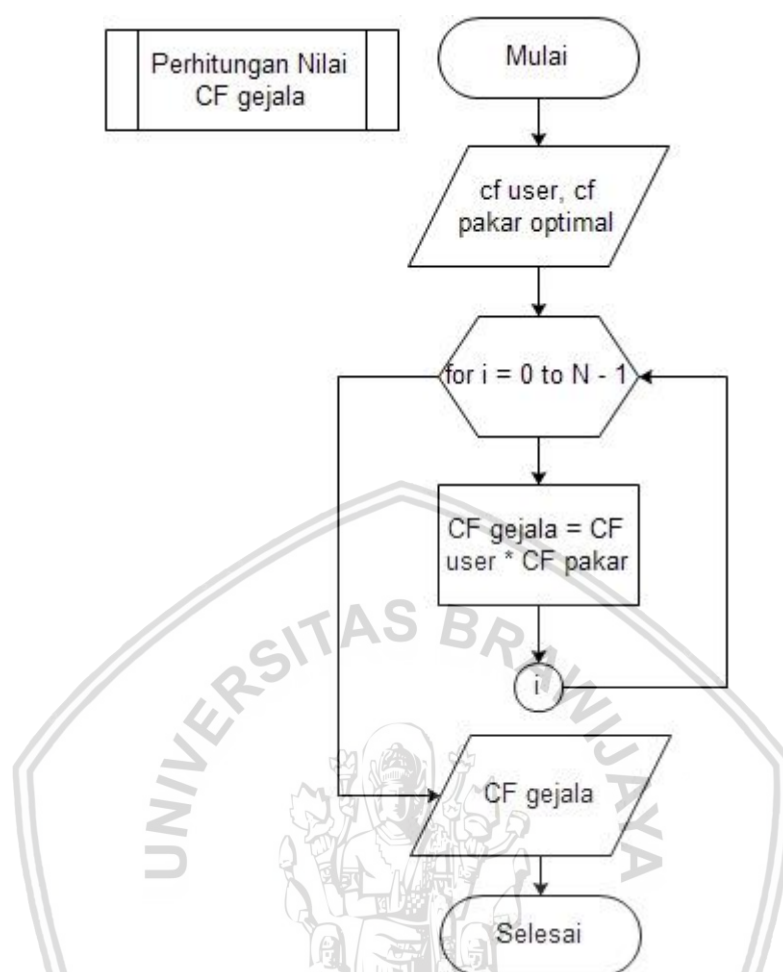
Langkah-langkah *update personal best* dan *global best* pada algoritma PSO berdasarkan Gambar 4.5 adalah sebagai berikut:

1. Sistem menerima masukan berupa *personal best*, *global best*, *cost* sebelumnya, *cost* sekarang.
2. Melakukan *update personal best* dengan membandingkan nilai *cost* sebelumnya dan nilai *cost* sekarang.
3. Melakukan *update global best* dengan mencari nilai *cost* terkecil dari *personal best*.
4. Keluaran sistem ini adalah *personal best* dan *global best* yang sudah diperbaharui.

#### 4.2.6 Perhitungan CF Gejala

Perhitungan Cf gejala bertujuan untuk mengetahui nilai kepercayaan dari gejala atau fakta tunggal terhadap suatu indikasi. Perhitungan Cf gejala dilakukan dengan perkalian nilai Cf pakar setelah dioptimasi menggunakan algoritma *Particle Swarm Optimization (PSO)* dengan Cf user yang sebelumnya telah diinputkan pengguna sesuai dengan Persamaan 2.2. Diagram alir proses perhitungan Cf gejala ditunjukkan pada Gambar 4.6 sebagai berikut:





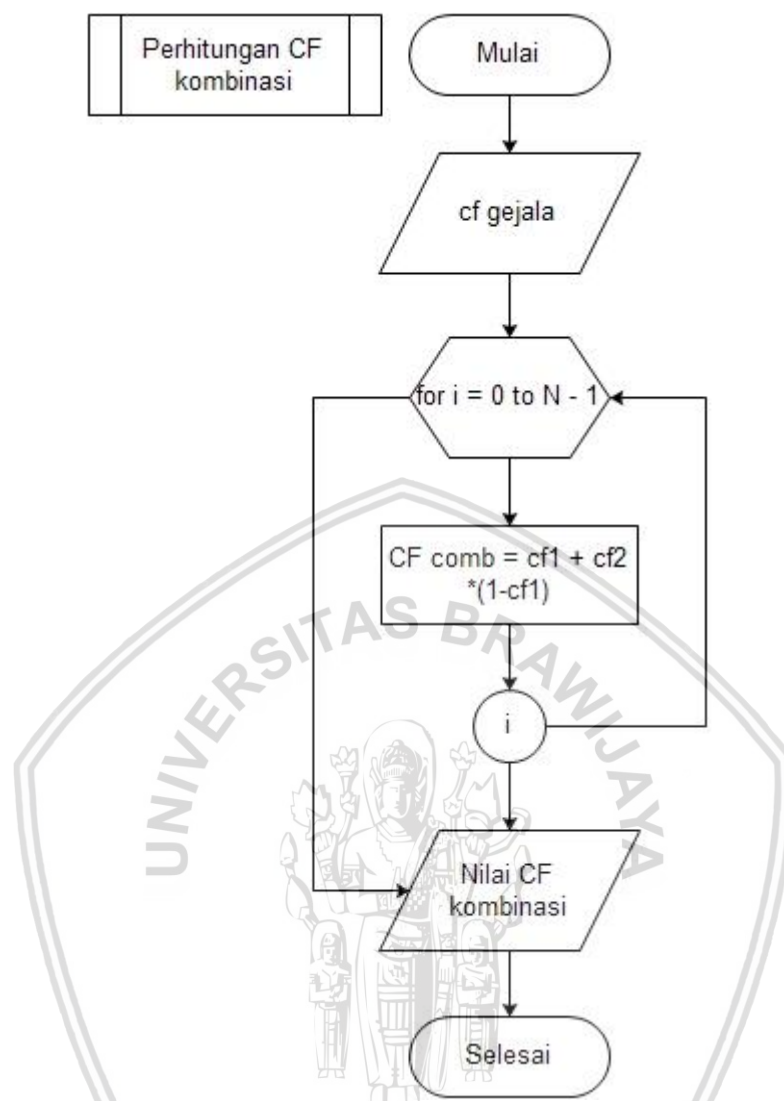
**Gambar 4.6 Diagram Alir Proses Perhitungan Cf Gejala**

Langkah-langkah proses perhitungan Cf gejala pada algoritma *Certainty factor* berdasarkan Gambar 4.6 adalah sebagai berikut:

1. Sistem menerima masukan berupa Cf user dan Cf pakar yang sebelumnya sudah dipotimasi dengan *PSO*.
2. Untuk gejala 1 sampai jumlah gejala yang diinputkan oleh *user* ulangi langkah ke 3.
3. Melakukan perhitungan Cf gejala dengan perkalian Cf user dan Cf pakar sesuai Persamaan 2.2.
4. Hasil akhir proses adalah nilai Cf gejala dari setiap fakta yang diinputkan oleh *user*.

#### 4.2.7 Perhitungan CF Kombinasi

Perhitungan nilai Cf kombinasi dilakukan dengan cara melakukan perhitungan kombinasi tiap nilai cf gejala yang telah didapat sebelumnya untuk mencari seberapa besar nilai tingkat keyakinan suatu indikasi yang dipengaruhi oleh gejala – gejala tersebut. Diagram alir proses perhitungan Cf kombinasi ditunjukkan pada Gambar 4.7 sebagai berikut:



**Gambar 4.7 Diagram Alir Proses Perhitungan Cf Kombinasi**

Langkah-langkah proses perhitungan cf kombinasi pada algoritma *Certainty factor* berdasarkan Gambar 4.7 adalah sebagai berikut:

1. Sistem menerima masukan berupa Cf gejala yang sebelumnya didapat.
2. Untuk Cf gejala 1 sampai jumlah gejala yang diinputkan oleh user ulangi langkah 3.
3. Melakukan perhitungan Cf kombinasi menggunakan Persamaan 2.3.
4. Hasil akhir proses adalah nilai Cf kombinasi dari beberapa Cf gejala.

#### 4.3 Siklus Penyelesaian Masalah Menggunakan Algoritma PSO dan Certainty Factor

Siklus algoritma *PSO* dan *Certainty Factor* yang telah diuraikan pada subbab 4.2 selanjutnya akan disederhanakan menjadi perhitungan manual untuk

memudahkan pemahaman tentang penyelesaian masalah sebelum diimplementasikan ke dalam kode program.

Perhitungan manual pada penelitian ini menggunakan sampel data bobot nilai Cf pakar yang dibangkitkan secara acak sebanyak jumlah partikel. Berikut merupakan inisialisasi awal yang digunakan dalam perhitungan manual:

- Jumlah iterasi : 1
- Ukuran swarm : 5
- Bobot inersia maksimal : 0.9
- Bobot inersia minimal : 0.4
- Koefisien akselerasi 1 & 2 : 2
- Maksimal & minimal kecepatan : 0,05 & -0,05
- Maksimal & minimal posisi : 1 & 0

#### 4.3.1 Inisialisasi Partikel

Sebagai contoh, telah ditentukan ukuran swarm sebanyak 5 partikel. Pada inisialisasi partikel nilai yang di acak adalah 0 (nol) sampai 1. Nilai dari tiap partikel merupakan nilai Cf pakar masing masing fakta yang telah dimasukan, sehingga inisialisasi partikel dibentuk menjadi matrik yang ditunjukan pada Tabel 4.5 sebagai berikut:

**Tabel 4.4 Nilai Inisialisasi Partikel**

P	A			B			C			D			E		
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T
1	0,34	0,15	0,26	0,43	0,12	0,19	0,54	0,67	0,18	0,28	0,08	0,41	0,13	0	0,23
2	0,5	0,07	0,13	0,31	0,07	0,11	0,42	0,5	0,13	0,26	0,05	0,43	0,09	0	0,16
3	0,31	0,1	0,24	0,46	0,17	0,28	0,4	0,55	0,06	0,13	0,17	0,56	0,005	0	0,28
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0	0,11
5	0,39	0,03	0,21	0,35	0,08	0,22	0,41	0,64	0,05	0,19	0,09	0,45	0,07	0	0,17

Keterangan :

- P = Partikel ke-
- K = Kolam cor beton
- S = Pemberian antiseptik 1-2 kali / minggu
- T = Air tampak keruh
- A = Fakta K,S,T Terhadap indikasi Terserang Parasit
- B = Fakta K,S,T Terhadap indikasi Terserang Bakteri
- C = Fakta K,S,T Terhadap indikasi Ikan Sehat
- D = Fakta K,S,T Terhadap indikasi Keracunan
- E = Fakta K,S,T Terhadap indikasi Ikan Mati

#### 4.3.2 Perhitungan Nilai Cost Iterasi 0

Setiap partikel akan menghasilkan nilai *cost* untuk menentukan seberapa optimal solusi yang didapatkan. Nilai *cost* bertujuan untuk meminimalkan jarak partikel terhadap nilai bobot. Ada 4 tahapan untuk mendapatkan nilai *cost* yaitu:

### 1. Nilai Partikel

Pada tahap ini, nilai partikel yang didapat sama seperti saat inisialisasi partikel. Nilai partikel ditunjukkan pada tabel 4.5.

**Tabel 4.5 Nilai Partikel**

P	A			B			C			D			E		
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T
1	0,34	0,15	0,26	0,43	0,12	0,19	0,54	0,67	0,18	0,28	0,08	0,41	0,13	0	0,23
2	0,5	0,07	0,13	0,31	0,07	0,11	0,42	0,5	0,13	0,26	0,05	0,43	0,09	0	0,16
3	0,31	0,1	0,24	0,46	0,17	0,28	0,4	0,55	0,06	0,13	0,17	0,56	0,05	0	0,28
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0	0,11
5	0,39	0,03	0,21	0,35	0,08	0,22	0,41	0,64	0,05	0,19	0,09	0,45	0,07	0	0,17

### 2. Inisialisasi Nilai Bobot Pakar

Pada tahap ini inisialisasi bobot pakar didapat dari hasil wawancara, maka diperoleh bobot pakar pada Tabel 4.6.

**Tabel 4.6 Nilai Bobot Pakar**

P	A	B	C	D	E
Kolam cor beton	0,45	0,39	0,58	0,27	0,18
Air tampak keruh	0,11	0,14	0,15	0,53	0,11
Pemberian antiseptik 1-2 kali/minggu	0,18	0,02	0,61	0,13	0

### 3. Menghitung Nilai Cost

Berdasarkan persamaan 2.7 maka diperoleh hasil nilai cost pada Tabel 4.7.

**Tabel 4.7 Hasil Nilai Cost**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
1	-0,26	-0,09	-0,15	-0,23	-0,09	-0,16	-0,23	-0,20	-0,08	-0,14	-0,09	-0,30	-0,09	0,00	-0,15	-2,26
2	-0,20	-0,09	-0,17	-0,28	-0,09	-0,18	-0,29	-0,30	-0,09	-0,15	-0,10	-0,29	-0,09	0,00	-0,17	-2,48
3	-0,28	-0,09	-0,15	-0,22	-0,08	-0,14	-0,30	-0,27	-0,09	-0,17	-0,08	-0,22	-0,10	0,00	-0,14	-2,34
4	-0,22	-0,08	-0,18	-0,24	-0,10	-0,17	-0,21	-0,23	-0,09	-0,15	-0,09	-0,24	-0,08	0,00	-0,18	-2,25
5	-0,24	-0,07	-0,16	-0,26	-0,09	-0,16	-0,30	-0,22	-0,10	-0,16	-0,09	-0,28	-0,09	0,00	-0,17	-2,37

### 4. Penentuan Nilai Cost Terbaik

Pada tahap ini, nilai cost berdasarkan nilai terkecil terhadap 0 (nol) dari jumlah nilai masing-masing partikel. Maka diperoleh nilai cost pada Tabel 4.8.

**Tabel 4.8 Nilai Cost Terbaik**

Partikel ke-	Nilai Cost
4	-2,25

#### 4.3.3 Perhitungan Inisialisasi Personal Best dan Global Best

Inisialisasi nilai *personal best* nilainya sama seperti inisialisasi partikel *PSO*. Nilai *personal best* ditunjukkan pada tabel 4.9.

**Tabel 4.9 Inisialisasi Personal Best**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
1	0,34	0,15	0,26	0,43	0,12	0,19	0,54	0,67	0,18	0,28	0,08	0,41	0,13	0	0,23	-2,26
2	0,5	0,07	0,13	0,31	0,07	0,11	0,42	0,5	0,13	0,26	0,05	0,43	0,09	0	0,16	-2,48
3	0,31	0,1	0,24	0,46	0,17	0,28	0,4	0,55	0,06	0,13	0,17	0,56	0,05	0	0,28	-2,34
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0	0,11	-2,25
5	0,39	0,03	0,21	0,35	0,08	0,22	0,41	0,64	0,05	0,19	0,09	0,45	0,07	0	0,17	-2,37

Sedangkan nilai *global best* merupakan nilai *personal best* yang mempunyai nilai *cost* terkecil. inisialisasi nilai *global best* ditunjukkan pada Tabel 4.10.

**Tabel 4.10 Inisialisasi Global Best**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0	0,11	-2,25

#### 4.3.4 Perhitungan Update Kecepatan dan Posisi

*Update* kecepatan dan posisi partikel pada iterasi 0 dan selanjutnya dihitung berdasarkan persamaan 2.8 dan 2.10. Hasil perhitungan kecepatan dari semua partikel ditunjukkan pada Tabel 4.11.

**Tabel 4.11 Perhitungan Update Kecepatan Partikel**

P	A			B			C			D			E		
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T
1	0,11	0,03	-0,14	-0,04	-0,10	-0,05	0,04	-0,06	-0,03	-0,01	0,05	0,11	0,05	0,00	-0,11
2	-0,05	0,11	-0,02	0,08	-0,05	0,03	0,15	0,11	0,02	0,01	0,08	0,10	0,09	0,00	-0,05
3	0,13	0,08	-0,12	-0,07	-0,14	-0,13	0,17	0,06	0,09	0,13	-0,04	-0,03	0,12	0,00	-0,16
4	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	0,06	-0,11	-0,10	0,04	-0,06	-0,08	0,16	-0,03	0,10	0,08	0,04	0,08	0,11	0,00	-0,06

Selanjutnya hasil perhitungan kecepatan dari semua partikel dikonversi dengan nilai -0.05, 0, dan 0.05 yang ditunjukkan pada Tabel 4.12.

**Tabel 4.12 Konversi Kecepatan Partikel**

P	A			B			C			D			E		
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T
1	0,05	0,05	-0,05	-0,05	-0,05	-0,05	0,05	-0,05	-0,05	-0,05	0,05	0,05	0,05	0	-0,05
2	-0,05	0,05	-0,05	0,05	-0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	-0,05
3	0,05	0,05	-0,05	-0,05	-0,05	-0,05	0,05	0,05	0,05	0,05	-0,05	-0,05	0,05	0	-0,05
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0,05	-0,05	-0,05	0,05	-0,05	-0,05	0,05	-0,05	0,05	0,05	0,05	0,05	0,05	0	-0,05

Setelah didapatkan kecepatan masing-masing partikel selanjutnya dihitung posisi partikel menggunakan persamaan 2.10. Hasil perhitungan posisi dari semua partikel ditunjukkan pada Tabel 4.13.

**Tabel 4.13 Perhitungan *Update* Posisi Partikel**

P	A			B			C			D			E		
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T
1	0,39	0,20	0,21	0,38	0,07	0,14	0,59	0,62	0,13	0,23	0,13	0,46	0,18	0,00	0,18
2	0,45	0,12	0,08	0,36	0,02	0,16	0,47	0,55	0,18	0,31	0,10	0,48	0,14	0,00	0,11
3	0,36	0,15	0,19	0,41	0,12	0,23	0,45	0,60	0,11	0,18	0,12	0,51	0,10	0,00	0,23
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0,00	0,11
5	0,44	0,25	0,16	0,40	0,03	0,17	0,46	0,59	0,10	0,24	0,14	0,50	0,12	0,00	0,12

#### 4.3.5 Perhitungan Nilai *Cost* Iterasi 1

Seperti pada iterasi 0, nilai *cost* pada iterasi 1 dihitung untuk masing-masing partikel. Hasil perhitungan nilai *cost* iterasi 1 ditunjukkan pada Tabel 4.14.

**Tabel 4.14 Perhitungan Nilai *Cost* Iterasi 1**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
1	-0,24	-0,08	-0,16	-0,25	-0,09	-0,17	-0,21	-0,23	-0,09	-0,15	-0,09	-0,27	-0,08	0,00	-0,16	-2,27
2	-0,22	-0,09	-0,18	-0,26	-0,10	-0,17	-0,27	-0,27	-0,08	-0,14	-0,09	-0,26	-0,09	0,00	-0,18	-2,38
3	-0,26	-0,09	-0,16	-0,24	-0,09	-0,15	-0,28	-0,24	-0,09	-0,16	-0,09	-0,25	-0,09	0,00	-0,15	-2,33
4	-0,22	-0,08	-0,18	-0,24	-0,10	-0,17	-0,21	-0,23	-0,09	-0,15	-0,09	-0,24	-0,08	0,00	-0,18	-2,25
5	0,18	0,03	0,03	0,16	0,00	0,03	0,23	0,35	0,01	0,05	0,01	0,25	0,01	0,00	0,02	1,37

#### 4.3.6 Perhitungan *Update* Personal Best dan Global Best

*Update personal best* pada iterasi 1 dan selanjutnya didapatkan dari nilai terkecil dengan membandingkan nilai *cost personal best* iterasi saat ini dengan nilai *cost personal best* pada iterasi sebelumnya. Sedangkan *global best* diperoleh



dari partikel dengan nilai *cost* terkecil dalam *personal best* yang telah terpilih. Tabel 4.15 menunjukkan perbandingan nilai *cost personal best* iterasi 0 dan iterasi 1.

**Tabel 4.15 Perbandingan Cost Iterasi 0 dan Iterasi 1**

Partikel	Cost Iterasi 0	Cost Iterasi 1
1	-2,26	-2,27
2	-2,48	-2,38
3	-2,34	-2,33
4	-2,25	-2,25
5	-2,37	1,37

Setelah dilakukan perbandingan dengan nilai *cost personal best* iterasi 0 maka dilakukan proses *update personal best*. Hasil *update personal best* ditunjukkan pada Tabel 4.16.

**Tabel 4.16 Update Personal Best Iterasi 1**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
1	0,39	0,20	0,21	0,38	0,07	0,14	0,59	0,62	0,13	0,23	0,13	0,46	0,18	0,00	0,18	-2,26
2	0,45	0,12	0,08	0,36	0,02	0,16	0,47	0,55	0,18	0,31	0,10	0,48	0,14	0,00	0,11	-2,38
3	0,36	0,15	0,19	0,41	0,12	0,23	0,45	0,60	0,11	0,18	0,12	0,51	0,10	0,00	0,23	-2,33
4	0,45	0,18	0,11	0,39	0,02	0,14	0,58	0,61	0,15	0,27	0,13	0,53	0,18	0,00	0,11	-2,25
5	0,44	0,25	0,16	0,40	0,03	0,17	0,46	0,59	0,10	0,24	0,14	0,50	0,12	0,00	0,12	1,37

Selanjutnya dilakukan *update* nilai *global best* berdasarkan nilai *cost* terkecil dari *personal best* yang telah diperbarui. Hasil *update global best* ditunjukkan pada Tabel 4.17.

**Tabel 4.17 Update Global Best Iterasi 1**

P	A			B			C			D			E			Cost
	K	S	T	K	S	T	K	S	T	K	S	T	K	S	T	
5	0,44	0,25	0,16	0,40	0,03	0,17	0,46	0,59	0,10	0,24	0,14	0,50	0,12	0,00	0,12	1,37

#### 4.3.7 Perhitungan CF Gejala

Memasuki perhitungan *Certainty factor*, nilai Cf pakar didapat dari hasil optimasi *PSO* ditunjukkan pada Tabel 4.18.

Tabel 4.17 Nilai CF Pakar Setelah Optimasi

Indikasi	F2	F13	F18
Terserang Bakteri (P1)	0,44	0,16	0,25
Terserang Parasit (P2)	0,40	0,17	0,03
Ikan Sehat (P3)	0,46	0,10	0,59
Keracunan (P4)	0,24	0,50	0,14
Ikan Mati (P5)	0,12	0,12	0,00

Pada kasus ini, perhitungan dengan masukan 3 fakta dari pengguna. Dalam kasus ini pengguna mengkombinasikan fakta Kolam Cor Beton (F2), Air Tampak Keruh (F13) dan Pemberian Antiseptik 1-2 kali/minggu (F18). Sedangkan tingkat kepercayaan user terhadap fakta yang terjadi ( $Cf_{user}$ ) berturut-turut adalah Yakin (1), Kurang yakin (0.4), dan Yakin (1) ditunjukkan pada tabel 4.18.

Tabel 4.18 Nilai CF User

Fakta	$Cf_{user}$
Kolam Cor Beton (F2)	1
Air Tampak Keruh (F13)	0.4
Pemberian Antiseptik 1-2 kali/minggu (F18)	1

Dengan kombinasi fakta yang dipilih oleh pengguna, maka *rule* yang terbentuk adalah sebagai berikut :

IF F2 THEN P1  
 IF F2 THEN P2  
 IF F2 THEN P3  
 IF F2 THEN P4  
 IF F2 THEN P5  
 IF F13 THEN P1  
 IF F13 THEN P2  
 IF F13 THEN P3  
 IF F13 THEN P4  
 IF F13 THEN P5  
 IF F18 THEN P1  
 IF F18 THEN P2  
 IF F18 THEN P3  
 IF F18 THEN P4

➤ **Fakta 1 : F2**

Dari *rule* diatas kita bisa menghitung nilai Cf gejala dari fakta kolam cor beton (F2) terhadap semua indikasi yaitu terserang bakteri (P1), terserang parasit (P2), ikan sehat (P3), keracunan (P4), dan mati (P5). Perhitungan Cf gejala menggunakan Persamaan 2.2 sebagai berikut.

1. Cf gejala F2 terhadap P1.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,44 \\ &= 0,44 \end{aligned}$$

2. Cf gejala F2 terhadap P2.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,40 \\ &= 0,40 \end{aligned}$$

3. Cf gejala F2 terhadap P3.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,46 \\ &= 0,46 \end{aligned}$$

4. Cf gejala F2 terhadap P4.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,24 \\ &= 0,24 \end{aligned}$$

5. Cf gejala F2 terhadap P5.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,12 \\ &= 0,12 \end{aligned}$$

➤ **Fakta 2 : F13**

Sama seperti fakta 1, menghitung nilai Cf gejala dari fakta air tampak keruh (F13) terhadap semua indikasi yaitu terserang bakteri (P1), terserang parasit (P2), ikan sehat (P3), keracunan (P4), dan mati (P5). Perhitungan Cf gejala menggunakan Persamaan 2.2 sebagai berikut.

1. Cf gejala F13 terhadap P1.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 0,4 \times 0,16 \\ &= 0,064 \end{aligned}$$

2. Cf gejala F13 terhadap P2.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 0,4 \times 0,17 \\ &= 0,068 \end{aligned}$$

3. Cf gejala F13 terhadap P3.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 0,4 \times 0,10 \\ &= 0,04 \end{aligned}$$

4. Cf gejala F13 terhadap P4.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 0,4 \times 0,50 \\ &= 0,2 \end{aligned}$$

5. Cf gejala F13 terhadap P5.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 0,4 \times 0,12 \\ &= 0,048 \end{aligned}$$

➤ **Fakta 3 : F18**

Menghitung nilai Cf gejala dari fakta pemberian antiseptik 1-2 kali/minggu (F18) terhadap semua indikasi yaitu terserang bakteri (P1), terserang parasit (P2), ikan sehat (P3), keracunan (P4), dan mati (P5). Perhitungan Cf gejala menggunakan Persamaan 2.2 sebagai berikut.

1. Cf gejala F18 terhadap P1.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,25 \\ &= 0,25 \end{aligned}$$

2. Cf gejala F18 terhadap P2.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,03 \\ &= 0,03 \end{aligned}$$

3. Cf gejala F18 terhadap P3.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,59 \\ &= 0,59 \end{aligned}$$

4. Cf gejala F18 terhadap P4.

$$\begin{aligned} Cf_{gejala} &= Cf_{user} \times Cf_{pakar} \\ &= 1 \times 0,14 \\ &= 0,14 \end{aligned}$$

Hasil dari perhitungan Cf gejala terhadap semua indikasi dapat dilihat pada tabel 4.19.

**Tabel 4.19 Hasil Perhitungan Cf Gejala**

Indikasi	CF Gejala		
	F2	F13	F18
Terserang Bakteri (P1)	0,44	0,064	0,25
Terserang Parasit (P2)	0,40	0,068	0,03
Ikan Sehat (P3)	0,46	0,04	0,59
Keracunan (P4)	0,24	0,2	0,14
Ikan Mati (P5)	0,12	0,048	0

#### 4.3.8 Perhitungan CF Kombinasi

Proses perhitungan Cf kombinasi dilakukan dengan cara melakukan perhitungan kombinasi setiap nilai Cf gejala yang sebelumnya telah didapat, yang bertujuan untuk mengetahui tingkat keyakinan semua indikasi yang dipengaruhi oleh gejala atau fakta - fakta tersebut. Perhitungan Cf kombinasi dilakukan menggunakan persamaan 2.3, dengan indikasi yang memiliki tingkat keyakinan paling tinggi sebagai hasil akhir.

##### ➤ Cf Kombinasi untuk P1

1. Kombinasi F2 & F13

$$\begin{aligned} Cf_{comb1} &= Cf_2 + Cf_{13} \times (1 - Cf_2) \\ &= 0,44 + 0,064 \times (1 - 0,44) \\ &= 0,475 \end{aligned}$$

2. Kombinasi  $Cf_{comb1}$  & F18

$$\begin{aligned} Cf_{comb2} &= Cf_{comb1} + Cf_{18} \times (1 - Cf_{comb1}) \\ &= 0,475 + 0,25 \times (1 - 0,475) \\ &= 0,60 \end{aligned}$$

Jadi Hasil Cf kombinasi untuk indikasi P1 sebesar 0,60.

##### ➤ Cf Kombinasi untuk P2

1. Kombinasi F2 & F13

$$\begin{aligned} Cf_{comb1} &= Cf_2 + Cf_{13} \times (1 - Cf_2) \\ &= 0,40 + 0,068 \times (1 - 0,40) \\ &= 0,44 \end{aligned}$$

3. Kombinasi  $Cf_{comb1}$  & F18

$$\begin{aligned} Cf_{comb2} &= Cf_{comb1} + Cf_{18} \times (1 - Cf_{comb1}) \\ &= 0,440 + 0,03 \times (1 - 0,440) \\ &= 0,45 \end{aligned}$$

Jadi Hasil Cf kombinasi untuk indikasi P2 sebesar 0,45.

##### ➤ Cf Kombinasi untuk P3

1. Kombinasi F2 & F13

$$\begin{aligned} Cf_{comb1} &= Cf_2 + Cf_{13} \times (1 - Cf_2) \\ &= 0,46 + 0,04 \times (1 - 0,46) \\ &= 0,48 \end{aligned}$$

4. Kombinasi  $Cf_{comb1}$  & F18

$$Cf_{comb2} = Cf_{comb1} + Cf_{18} \times (1 - Cf_{comb1})$$

$$= 0,481 + 0,59 \times (1 - 0,481) \\ = 0,79$$

Jadi Hasil Cf kombinasi untuk indikasi P3 sebesar 0,79.

➤ **Cf Kombinasi untuk P4**

2. Kombinasi F2 & F13

$$Cf_{comb1} = Cf_2 + Cf_{13} \times (1 - Cf_2) \\ = 0,24 + 0,2 \times (1 - 0,24) \\ = 0,39$$

5. Kombinasi  $Cf_{comb1}$  & F18

$$Cf_{comb2} = Cf_{comb1} + Cf_{18} \times (1 - Cf_{comb1}) \\ = 0,39 + 0,14 \times (1 - 0,39) \\ = 0,47$$

Jadi Hasil Cf kombinasi untuk indikasi P4 sebesar 0,47.

➤ **Cf Kombinasi untuk P5**

3. Kombinasi F2 & F13

$$Cf_{comb1} = Cf_2 + Cf_{13} \times (1 - Cf_2) \\ = 0,12 + 0,048 \times (1 - 0,12) \\ = 0,16$$

6. Kombinasi  $Cf_{comb1}$  & F18

$$Cf_{comb2} = Cf_{comb1} + Cf_{18} \times (1 - Cf_{comb1}) \\ = 0,16 + 0 \times (1 - 1,6) \\ = 0,16$$

Jadi Hasil Cf kombinasi untuk indikasi P5 sebesar 0,16.

Hasil dari perhitungan Cf kombinasi terhadap semua indikasi dapat dilihat pada tabel 4.20.

**Tabel 4.20 Hasil Perhitungan Cf Kombinasi**

CF Kombinasi	Indikasi				
	P1	P2	P3	P4	P5
	0,60	0,45	0,79	0,47	0,16

Berdasarkan hasil perhitungan dari Cf kombinasi maka diperoleh nilai Cf kombinasi paling tinggi adalah pada indikasi Ikan Sehat (P3) dengan nilai 0,79.



#### 4.4 Perancangan Pengujian Algoritma

Pengujian yang akan dilakukan pada penelitian ini yaitu pengujian terkait parameter algoritma *Particle Swarm Optimization* (PSO) yang digunakan untuk mengoptimasi nilai  $C_f$  pakar dan akurasi sistem. Skenario pengujian yang dilakukan antara lain:

1. Pengujian parameter *PSO*.
2. Pengujian akurasi sistem.

##### 4.4.2 Pengujian Parameter PSO

Pengujian ini dilakukan untuk mengetahui parameter algoritma *PSO* yang paling optimal dalam menentukan nilai  $C_f$  pakar. Untuk mendapatkan parameter *PSO* yang tepat, maka dilakukan uji coba sebagai berikut:

1. Pengujian bobot inersia
2. Pengujian koefisien akselerasi
3. Pengujian jumlah iterasi
4. Pengujian ukuran swarm

##### 4.4.2.1 Pengujian Bobot Inersia

Pengujian bobot inersia dilakukan untuk mengetahui kombinasi bobot inersia maksimal dan bobot inersia minimal yang tepat untuk menghasilkan nilai  $C_f$  pakar yang optimal. Bobot inersia maksimal dan minimal diuji coba pada rentang 0 sampai dengan 1 yang ditunjukkan pada Tabel 4.21. Parameter yang digunakan untuk pengujian bobot inersia adalah:

- a. Bobot inersia max : 0 - 1
- b. Bobot inersia min : 0 - 1
- c. Koefisien akselerasi 1 : 1
- d. Koefisien akselerasi 2 : 1
- e. Ukuran swarm : 10
- f. Jumlah iterasi : 100

**Tabel 4.21 Rancangan Pengujian Bobot Inersia**

Bobot Inersia		Rata-rata nilai Cost terbaik
$\omega_{max}$	$\omega_{min}$	
0,9	0,2	
	0,3	
	0,4	
0,8	0,2	
	0,3	
	0,4	
0,7	0,2	

	0,3	
	0,4	

#### 4.4.2.2 Pengujian Koefisien Akselerasi

Pengujian koefisien akselerasi dilakukan untuk mengetahui kombinasi koefisien akselerasi 1 dan koefisien akselerasi 2 yang tepat untuk menghasilkan nilai  $C_f$  pakar yang optimal. Koefisien akselerasi 1 dan 2 diuji coba pada rentang 1 sampai dengan 2 yang ditunjukkan pada Tabel 4.22. Parameter yang digunakan untuk pengujian koefisien akselerasi adalah:

- Bobot inersia max =  $\omega_{max}$  terbaik pada pengujian bobot inersia.
- Bobot inersia min =  $\omega_{min}$  terbaik pada pengujian bobot inersia.
- Koefisien akselerasi 1 = 1 - 2
- Koefisien akselerasi 2 = 1 - 2
- Ukuran swarm = 10
- Jumlah iterasi = 100

**Tabel 4.22 Rancangan Pengujian Koefisien Akselerasi**

Koefisien Akselerasi		Rata-rata nilai
$C_1$	$C_2$	Cost terbaik
1	1	
	1,2	
	2	
1,5	1	
	1,2	
	2	
2	1	
	1,2	
	2	

#### 4.4.2.3 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi dilakukan untuk mengetahui jumlah iterasi yang tepat dalam menentukan nilai  $C_f$  pakar yang optimal. Jumlah iterasi yang diujikan adalah kelipatan 100 yang ditunjukkan pada Tabel 4.23. Parameter yang digunakan pada pengujian jumlah iterasi adalah:

- Bobot inersia max =  $\omega_{max}$  terbaik pada pengujian bobot inersia
- Bobot inersia min =  $\omega_{min}$  terbaik pada pengujian bobot inersia
- Koefisien akselerasi 1 =  $C_1$  terbaik pada pengujian koefisien akselerasi

- d. Koefisien akselerasi 2 =  $C_2$  terbaik pada pengujian koefisien akselerasi  
 e. Ukuran swarm = 10  
 f. Jumlah iterasi = 100 – 500

Tabel 4.23 Rancangan Pengujian Jumlah Iterasi

Uji coba ke	Nilai cost terbaik				
	Iterasi = 100	Iterasi = 200	Iterasi = 300	Iterasi = 400	Iterasi = 500
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
Rata-rata					

#### 4.4.2.4 Pengujian Ukuran Swarm

Pengujian ukuran swarm dilakukan untuk mengetahui jumlah partikel yang dibutuhkan dalam menentukan nilai  $cf$  pakar yang optimal. Ukuran swarm yang diujikan adalah kelipatan 10 yang ditunjukkan pada Tabel 4.24. Parameter yang digunakan pada pengujian jumlah iterasi adalah:

- a. Bobot inersia max =  $\omega_{max}$  terbaik pada pengujian bobot inersia  
 b. Bobot inersia min =  $\omega_{min}$  terbaik pada pengujian bobot inersia  
 c. Koefisien akselerasi 1 =  $C_1$  terbaik pada pengujian koefisien akselerasi  
 d. Koefisien akselerasi 2 =  $C_2$  terbaik pada pengujian koefisien akselerasi  
 e. Ukuran swarm = 10 - 100  
 f. Jumlah iterasi = Jumlah terbaik pada pengujian jumlah iterasi

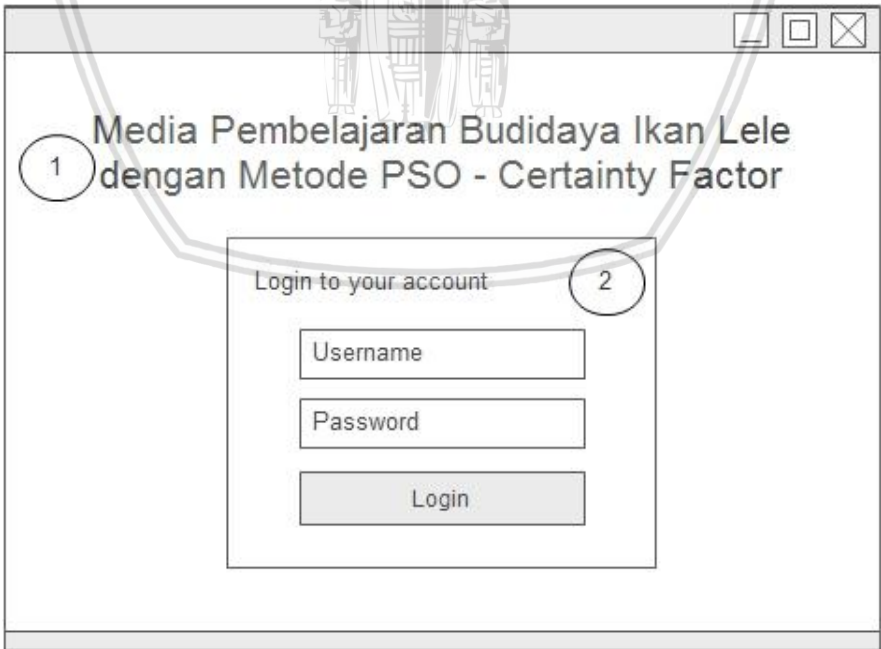
Tabel 4.24 Rancangan Pengujian Ukuran Swarm

Uji coba ke	Nilai cost terbaik				
	Partikel = 10	Partikel = 20	Partikel = ...	Partikel = 90	Partikel = 100
1					
2					

3					
4					
5					
6					
7					
8					
9					
10					
Rata-rata					

#### 4.5 Perancangan Antarmuka

Perancangan antarmuka digunakan untuk menggambarkan bagaimana implementasi algoritma *PSO-Certainty factor* untuk mengenali kondisi ikan lele yang akan dibangun. Antarmuka yang akan dirancang terdiri dari 9 antarmuka yaitu antarmuka login, antarmuka user diantaranya home, budidaya, hasil optimasi pso, hasil percobaan, dan kuis. Sedangkan pada antarmuka admin yaitu antarmuka home, parameter admin dan perhitungan cf. Antarmuka login dapat dilihat pada gambar 4.8.



Gambar 4.8 Perancangan Antarmuka Login

Keterangan perancangan antarmuka login yang ditunjukkan pada gambar 4.8 adalah sebagai berikut :

1. Merupakan judul dari sistem dengan latar belakang gambar pembudidayaan.
2. Form login yang terdapat *textfield* untuk memasukkan *username*, *textfield* untuk memasukkan *password*, serta tombol *login* untuk memproses masuk ke sistem.

#### 4.5.1 Perancangan Antarmuka User

Perancangan antarmuka halaman user dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.9.



**Gambar 4.9 Perancangan Antarmuka User Home**

Perancangan antarmuka user home yang ditunjukkan pada gambar 4.9 memiliki keterangan sebagai berikut :

1. Menu fitur yang ada di sistem diantaranya terdapat Home, Budidaya, Hasil Optimasi PSO, Hasil Percobaan, serta Kuis.
2. Konten berupa ucapan selamat datang di sistem bagi pengguna.
3. Menu sistem yang dapat digunakan *user* untuk *logout*.

#### 4.5.2 Perancangan Antarmuka Budidaya

Perancangan antarmuka halaman budidaya dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.10.



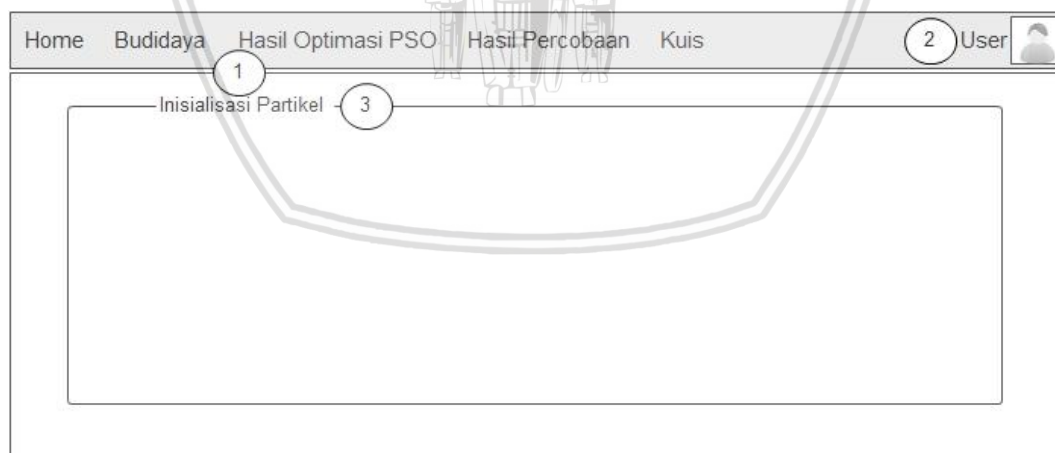
**Gambar 4.10 Perancangan Antarmuka Budidaya**

Keterangan perancangan antarmuka budidaya yang ditunjukkan pada gambar 4.10 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Budidaya, Hasil Optimasi PSO, Hasil Percobaan dan Kuis.
2. Menu *user* untuk *logout*.
3. Konten yang berisi fakta budidaya sebanyak 40 fakta dalam bentuk textbox.
4. Tombol mulai digunakan untuk memulai sistem.

#### 4.5.3 Perancangan Antarmuka Hasil Optimasi PSO

Perancangan antarmuka user hasil optimasi pso dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.11.



**Gambar 4.11 Perancangan Antarmuka Hasil Optimasi PSO**

Keterangan perancangan antarmuka hasil optimasi pso yang ditunjukkan pada gambar 4.11 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Budidaya, Hasil Optimasi PSO, Hasil Percobaan dan Kuis.
2. Menu *user* untuk *logout*.



3. Konten yang berisi hasil optimasi sistem.

#### 4.5.4 Perancangan Antarmuka Hasil Percobaan

Perancangan antarmuka user hasil percobaan dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.12.

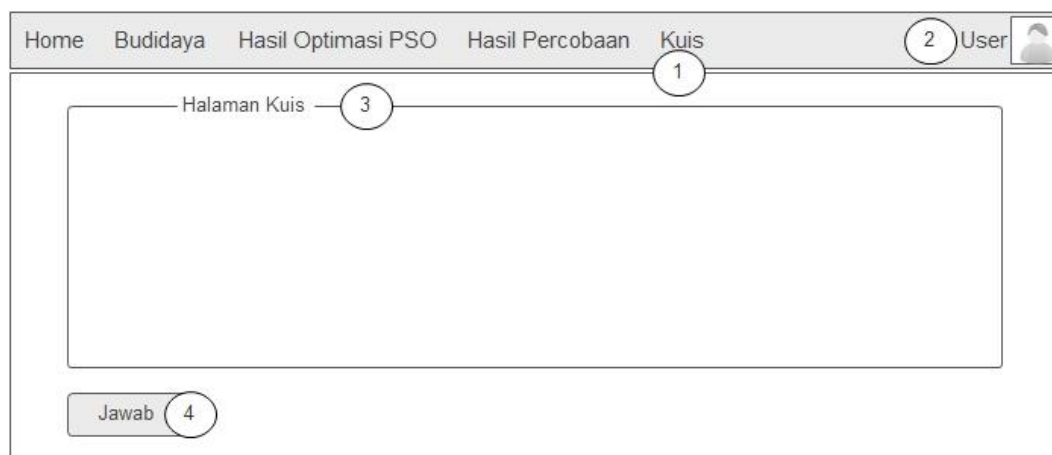
**Gambar 4.12 Perancangan Antarmuka Hasil Percobaan**

Keterangan perancangan antarmuka hasil percobaan yang ditunjukkan pada gambar 4.12 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Budidaya, Hasil Optimasi PSO, Hasil Percobaan dan Kuis.
2. Menu *user* untuk *logout*.
3. Konten yang berisi masukkan nilai Cf user.
4. Konten yang berisi hasil perhitungan *Certainty factor*.
5. Konten berisi kesimpulan hasil akhir sistem.

#### 4.5.5 Perancangan Antarmuka Kuis

Perancangan antarmuka user kuis dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.13.



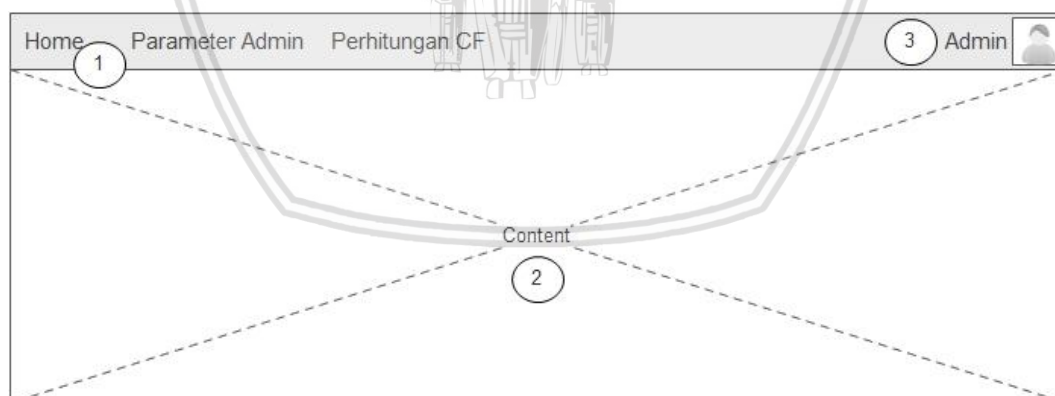
**Gambar 4.13 Perancangan Antarmuka Kuis**

Keterangan perancangan antarmuka kuis yang ditunjukkan pada gambar 4.13 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Budaya, Hasil Optimasi PSO, Hasil Percobaan dan Kuis.
2. Menu *user* untuk *logout*.
3. Konten yang berisi soal kuis.
4. Tombol jawab digunakan setelah selesai menjawab soal.

#### 4.5.6 Perancangan Antarmuka Admin Home

Perancangan antarmuka admin home dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.14.



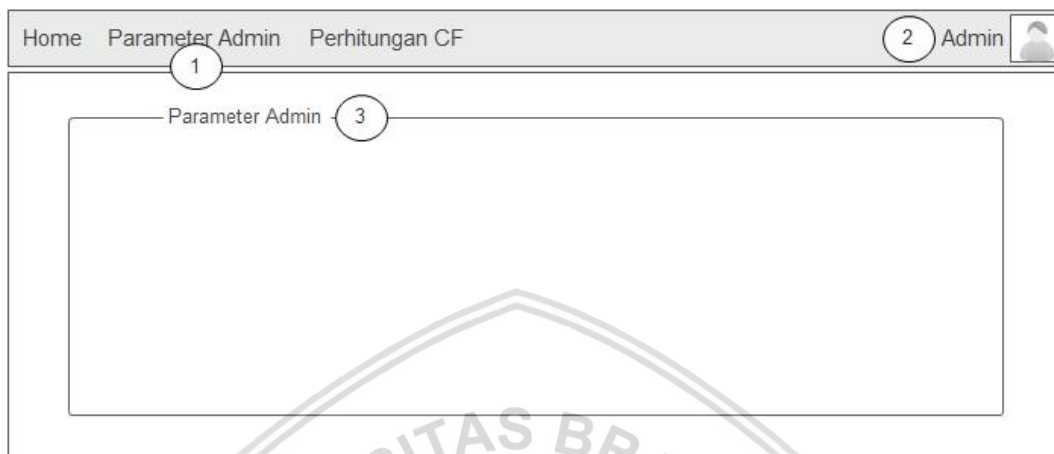
**Gambar 4.14 Perancangan Antarmuka Admin Home**

Keterangan perancangan antarmuka admin home yang ditunjukkan pada gambar 4.14 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Parameter Admin, dan Perhitungan CF.
2. Konten yang berisi ucapan selamat datang untuk admin.
3. Menu admin untuk *logout*.

#### 4.5.7 Perancangan Antarmuka Parameter Admin

Perancangan antarmuka parameter admin dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.15.



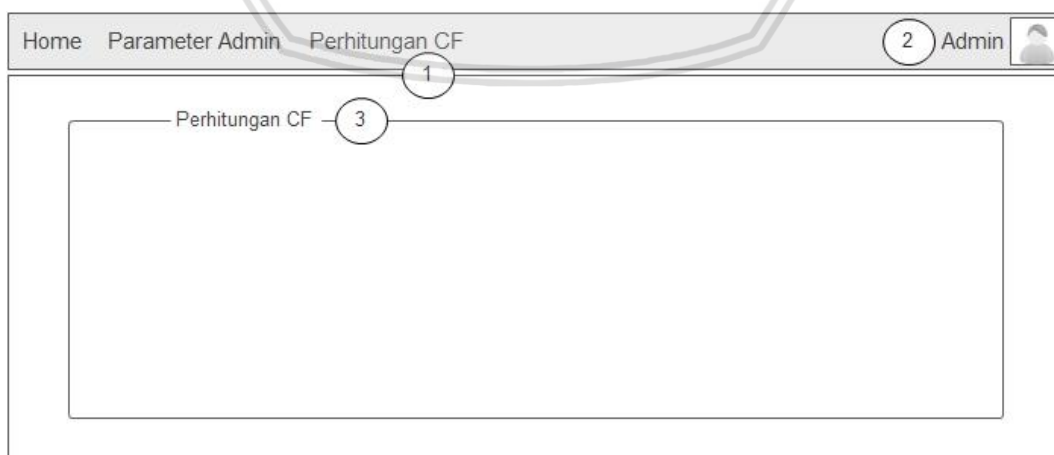
**Gambar 4.15 Perancangan Antarmuka Parameter Admin**

Keterangan perancangan antarmuka parameter admin yang ditunjukkan pada gambar 4.15 adalah sebagai berikut :

1. Menu sistem yang meliputi Home, Parameter Admin, dan Perhitungan CF.
2. Menu admin untuk *logout*.
3. Konten yang berfungsi untuk merubah parameter PSO.

#### 4.5.8 Perancangan Antarmuka Perhitungan CF

Perancangan antarmuka perhitungan cf dari implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele dapat dilihat pada Gambar 4.16.



**Gambar 4.16 Perancangan Antarmuka Perhitungan CF**

Keterangan perancangan antarmuka perhitungan cf yang ditunjukkan pada gambar 4.16 adalah sebagai berikut :

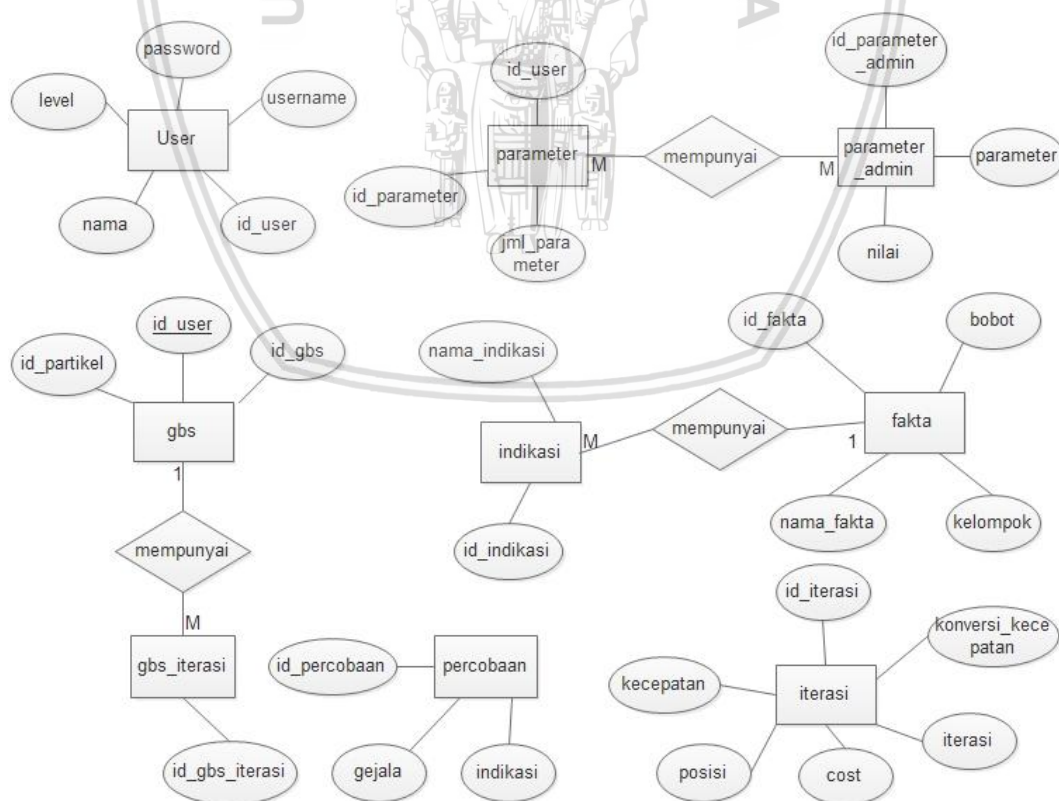
1. Menu sistem yang meliputi Home, Parameter Admin, dan Perhitungan CF.
2. Menu admin untuk *logout*.
3. Konten yang berisi proses perhitungan cf.

## 4.6 Perancangan Database

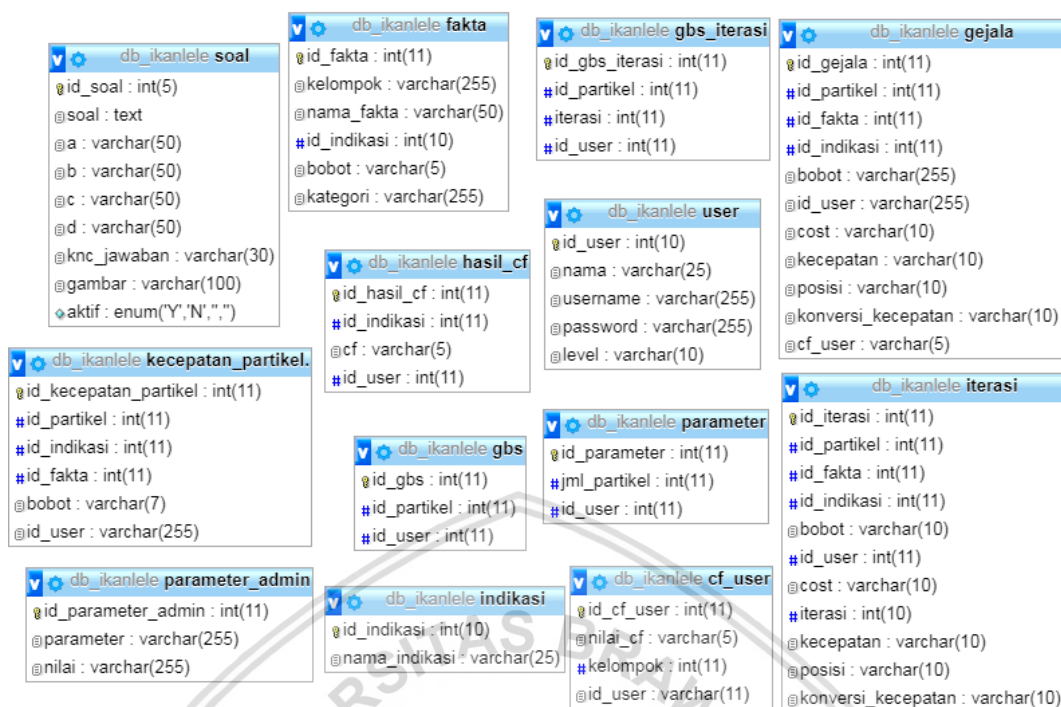
Perancangan database digunakan untuk menggambarkan semua database yang akan diimplementasikan pada implementasi metode *PSO-Certainty factor* untuk mengenali kondisi ikan lele. Perancangan database digambarkan dengan Entity Relationship Diagram (ERD).

### 4.6.1 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah diagram yang dipakai untuk mendokumentasikan data dengan mengidentifikasi jenis entitas dan hubungannya. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang merepresentasikan seluruh fakta yang ditinjau dari kenyataan nyata. Pada ERD sistem terdapat 9 entitas, yaitu entitas user, entitas gbs, entitas parameter, entitas parameter\_admin, entitas gbs\_iterasi, entitas percobaan, entitas iterasi, entitas fakta, dan entitas indikasi. Rancangan ERD sistem ditunjukkan pada Gambar 4.17 dan *Physical Diagram* ditunjukkan pada Gambar 4.18.



Gambar 4.17 Entity Relationship Diagram



Gambar 4.18 Physical Data Model

## BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem berdasarkan analisis kebutuhan dan proses perancangan sistem yang telah dibuat. Pembahasan pada bab ini terdiri dari penjelasan implementasi algoritma *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele serta implementasi antarmuka sistem.

### 5.1 Implementasi Algoritma Hybrid Particle Swarm Optimization dan Certainty Factor

Implementasi algoritma merupakan hasil perancangan implementasi metode *Particle Swarm Optimization-Certainty Factor* untuk mengenali kondisi ikan lele ke dalam kode program. Implementasi sistem berbasis web dengan bahasa pemrograman PHP. Sistem ini terdiri dari 8 proses meliputi proses inisialisasi partikel, proses menghitung nilai cost, proses update personal best, proses update global best, proses update kecepatan dan posisi, proses inisialisasi nilai cf pakar, proses menghitung nilai cf gejala, proses menghitung cf kombinasi.

#### 5.1.1 Implementasi Inisialisasi Partikel

Implementasi inisialisasi partikel merupakan proses awal untuk menginisialisasi algoritma *PSO*. Proses ini diawali dengan membangkitkan nilai cf pakar secara acak 0 (nol) sampai dengan 1 sebanyak fakta yang dipilih pengguna. Implementasi inisialisasi partikel ditunjukkan pada Gambar 5.1

```

1. for ($i = 1; $i <= $jml_partikel; $i++) {
2.   $cost = 0;
3.   $cf = 0;
4.   foreach ($id_fakta as $idfakta) {
5.     $sqlfkt = "select * from fakta WHERE id_fakta
6.     ='$idfakta'";
7.     $exec = mysql_query($sqlfkt) or die(mysql_error());
8.     $detfkt = mysql_fetch_array($exec);
9.     //cek bobot lebih dari 0
10.    if ($detfkt['bobot'] > 0) {
11.      //perhitungan bobot
12.      //$bobot = rand($random_min * 100, $random_max * 100) /
13.      100;
14.      $bobot = rand((( $detfkt['bobot'] * 100) - 20),
15.      (( $detfkt['bobot'] * 100) + 20)) / 100;
16.    } else {
17.      $bobot = 0;
18.    }

```

Gambar 5.1 Kode Program Inisialisasi Partikel



Penjelasan Gambar 5.1 tentang implementasi inisialisasi partikel adalah:

Baris 1 s.d 8 : Mengecek inputan fakta dari user.

Baris 9 s.d 14 : Mengecek nilai bobot bernilai 0 atau tidak. Jika bernilai 0 maka tidak dibangkitkan nilai random dan sebaliknya jika tidak bernilai 0 maka akan di bangkitkan

### 5.1.2 Implementasi Menghitung Nilai Cost

Implementasi menghitung *cost* merupakan proses untuk menghitung nilai *cost* setiap partikel. Nilai *cost* diperoleh dengan mengkalikan nilai bobot dengan tiap partikel dan mengurangnya dengan nilai bobot seperti pada Persamaan 2.7. Dengan perhitungan itu, maka *cost* yang paling mendekati 0 (nol) adalah yang dipilih. Kode program menghitung nilai *cost* ditunjukkan pada Gambar 5.2.

```

1. foreach ($id_fakta as $idfakta) {
2.   $sqlfkt = "select * from fakta WHERE id_fakta =
3.   '$idfakta'";
4.   $exec = mysql_query($sqlfkt) or die(mysql_error());
5.   $detfkt = mysql_fetch_array($exec);
6. //cek bobot lebih dari 0
7.   if ($detfkt['bobot'] > 0) {
8. //perhitungan bobot
9. //   $bobot = rand($random_min * 100, $random_max * 100)
10.    / 100;
11.     $bobot = rand((( $detfkt['bobot'] * 100) - 20),
12.     (( $detfkt['bobot'] * 100) + 20)) / 100;
13.   } else {
14.     $bobot = 0;
15.   }
16.   $cost += ($bobot * $detfkt['bobot']) -
17.   $detfkt['bobot'];
18.   mysql_query("insert into gejala VALUES (NULL
19.   , '$i', '$idfakta', '$detfkt[id_indikasi]', '$bobot',
20.   '$_SESSION
21.   id]', '', '', '', '', '$cf_user[$cf]')") or
22.   die(mysql_error());
23.   $cf++;
24.   }

```

**Gambar 5.2 Kode Program Inisialisasi Menghitung Nilai Cost**

Penjelasan Gambar 5.2 tentang implementasi menghitung nilai *cost* adalah:

Baris 1 s.d 15 : Inisialisasi partikel

Baris 17 : Perhitungan nilai *cost* setiap partikel

Baris 18 s.d 24: Menyimpan hasil dari perhitungan *cost* ke database

### 5.1.3 Implementasi Personal Best

Implementasi *personal best* merupakan proses untuk menginisialisasi *personal best*. Pada proses ini untuk iterasi pertama, *personal best* diperoleh dari inisialisasi partikel, dan untuk iterasi selanjutnya akan diperbarui seiring *update* kecepatan dan posisi. Proses implementasi *personal best* ditunjukkan pada Gambar 5.3.

```

1. for ($i = 1; $i <= $jml_partikel; $i++) {
2.   $cost = 0; $cf = 0;
3.   foreach ($id_fakta as $idfakta) {
4.     $sqlfkt = "select * from fakta WHERE id_fakta =
5. '$idfakta'";
6.     $exec = mysql_query($sqlfkt) or die(mysql_error());
7.     $detfkt = mysql_fetch_array($exec);

```

**Gambar 5.3 Kode Program Personal Best**

Penjelasan Gambar 5.3 tentang implementasi personal best adalah:

Baris 1 s.d 2 : Perulangan untuk menampilkan seluruh partikel sebanyak jumlah partikel

Baris 3 s.d 6 : Mengambil data dari database fakta

### 5.1.4 Implementasi Global Best

Implementasi *global best* merupakan proses untuk menginisialisasi *global best*. Nilai *global best* diperoleh dari *personal best* berdasarkan nilai *cost* paling mendekati 0 (nol). Proses implementasi inisialisasi *global best* ditunjukkan pada Gambar 5.4.

```

1. if ($cost_pil < 0) {
2.   array_push($negatif_ite, $cost_pil);
3. } else {
4.   array_push($positif_ite, $cost_pil);
5. }
6. }
7. if (count($positif_ite) > 0 && count($negatif_ite) > 0)
8. {
9.   $gbs = abs(max($negatif_ite)) <
10.  abs(min($positif_ite)) ?
11.  max($negatif_ite) : min($positif_ite);
12.   } elseif (count($positif_ite) > 0 &&
13.  count($negatif_ite) <= 0) {
14.   $gbs = min($positif_ite);
15. } elseif (count($positif_ite) <= 0 &&
16.  count($negatif_ite)
17.  > 0) {
18.   $gbs = max($negatif_ite);
19. }
20. $sql_det = "select * from iterasi where iterasi =
21. '$i' and
22. cost = '$gbs' and id_user = '$_SESSION[id]'";
23. $exec_det = mysql_query($sql_det) or
24. die(mysql_error());

```

```

25. $det = mysql_fetch_array($exec_det);
26. mysql_query("DELETE FROM gbs_iterasi WHERE iterasi =
27. '$i'
28. AND id_user = '$_SESSION[id]') or
29. die(mysql_error());
30. mysql_query("INSERT INTO gbs_iterasi VALUES
31. ('','$det[id_partikel]','$i','$_SESSION[id]')") or
32. die(mysql_error());
33. $sql_itel = "select * from iterasi WHERE iterasi =
34. '$i' AND id_user = '$_SESSION[id]'";
35. $exec_itel = mysql_query($sql_itel) or
36. die(mysql_error());
37. $sql_gbs1 = "select * from gbs_iterasi
38. WHERE iterasi = '$i' AND id_user = '$_SESSION[id]'";
39. $exec_gbs1 = mysql_query($sql_itel) or
40. die(mysql_error());
41. $gbs1 = mysql_fetch_array($exec_gbs1);

```

**Gambar 5.4 Kode Program Global Best**

Penjelasan Gambar 5.4 tentang implementasi global best adalah:

- Baris 1 s.d 6 : Memeriksa nilai cost bernilai lebih dari 1 atau kurang dari 1
- Baris 7 s.d 19 : Menentukan Global best dengan memilih cost mendekati 0
- Baris 20 s.d 32: Proses update global best
- Baris 33 s.d 41: Memanggil global best baru setelah diupdate

### 5.1.5 Implementasi Update Kecepatan dan Posisi Partikel

Implementasi *update* kecepatan dan posisi partikel merupakan proses untuk memperbarui nilai kecepatan dan posisi partikel. Nilai kecepatan baru dihitung berdasarkan rumus Persamaan 2.8 sedangkan nilai posisi baru dihitung berdasarkan Persamaan 2.10. Implementasi update kecepatan dan posisi partikel ditunjukkan pada Gambar 5.5 sebagai berikut:

```

1. if ($data_itel['bobot'] == 0) {
2.   $kec = 0;
3. } else {
4.   $kec = (($bobot_inersia *
5.   $kecepatan_inisial) + (($c1 * $rand1) *
6.   $data_itel['bobot'] - $data_itel['bobot'])) + (($c2 *
7.   $rand2) - ($data_par['bobot'] -
8.   $data_itel['bobot']));
9.   }
10. //konversi kecepatan
11. if ($kec > 0) {
12.   $kon_kec =
13.   $data_itel['bobot'] + 0.005;
14. } elseif ($kec < 0) {
15.   $kon_kec =
16.   $data_itel['bobot'] + (-0.005);
17. } else {
18.   $kon_kec = 0;
19. }

```

```

20. mysql_query("update iterasi set
21. kecepatan = '" . $kec . "',posisi = '" . ($kec +
22. $data_itel['bobot']) . "',konversi_kecepatan =
23. '" . $kon_kec . "' where id_iterasi
24. ='$data_itel[id_iterasi]' and iterasi = '$i' and
25. id_user = '$_SESSION[id]') or
26. die(mysql_error());

```

**Gambar 5.5 Kode Program Update Kecepatan dan Posisi Partikel**

Penjelasan Gambar 5.5 tentang implementasi update kecepatan dan posisi partikel adalah:

Baris 1 s.d 2 : Memeriksa partikel dengan ketentuan jika bernilai 0 maka kecepatan bernilai 0

Baris 3 s.d 9 : Menghitung kecepatan partikel berdasarkan Persamaan 2.8

Baris 10 s.d 19: Konversi nilai kecepatan partikel.

Baris 20 s.d 26: Update kecepatan dan posisi pada database

### 5.1.6 Implementasi Inisialisasi CF Pakar

Implementasi nilai cf pakar merupakan proses awal dari algoritma *certainty factor*. Nilai cf pakar menunjukkan tingkat kepercayaan fakta terhadap indikasi yang sudah dioptimasi. Implementasi nilai cf pakar ditunjukkan pada Gambar 5.6 sebagai berikut:

```

1. <?php
2. $sql = "select * from cf_user WHERE id_user =
3. '$_SESSION[id]'";
4. $exec = mysql_query($sql) or die(mysql_error());
5. while ($data = mysql_fetch_array($exec)) {
6.     $par = fakta::bobotBykelompok($data['kelompok'], 1);
7.     $bobot = gejala::detailiterasifakta($par['id_fakta'],
8. 1);
9. ?>
10. <?php echo $bobot['bobot'] . "</br>";
11.     }
12. ?>

```

**Gambar 5.6 Kode Program Inisialisasi CF Pakar**

Penjelasan Gambar 5.6 tentang implementasi inisialisasi Cf pakar adalah:

Baris 1 s.d 4 : Mengambil data dari cf user

Baris 5 s.d 9 : Mengambil nilai cf pakar yang telah dioptimasi dari database

Baris 10 s.d 12: Menampilkan cf pakar setelah dioptimasi

### 5.1.7 Implementasi Perhitungan CF Gejala

Implementasi nilai cf gejala merupakan proses dari algoritma *certainty factor*. Implementasi nilai cf gejala ditunjukkan pada Gambar 5.7 sebagai berikut:

```

<?php
$i = 1;
$sql = "select * from cf_user WHERE id_user =

```

```
'$SESSION[id]''";
$exec = mysql_query($sql) or die(mysql_error());
while ($data = mysql_fetch_array($exec)) {
$par = fakta::bobotBykelompok($data['kelompok'], 1);
$bobot = gejala::detailiterasifakta($par['id_fakta'],
1);
if ($i == 1) {
$cf = $bobot['bobot'] * $data['nilai_cf'];
```

**Gambar 5.7 Kode Program Perhitungan CF Gejala**

Penjelasan Gambar 5.7 tentang implementasi perhitungan Cf gejala adalah:

Baris 1 s.d 3 : Mengambil nilai cf user

Baris 4 s.d 9 : Mengambil nilai cf pakar yang telah dioptimasi dari database

Baris 10 s.d 11: Menghitung nilai cf gejala

### 5.1.8 Implementasi Perhitungan CF Kombinasi

Implementasi nilai cf kombinasi merupakan proses lanjutan setelah didapat nilai cf gejala. Implementasi nilai cf kombinasi ditunjukkan pada Gambar 5.8 sebagai berikut:

```
<?php
$i = 1;
$sql = "select * from cf_user WHERE id_user =
'$SESSION[id]''";
$exec = mysql_query($sql) or die(mysql_error());
while ($data = mysql_fetch_array($exec)) {
$par = fakta::bobotBykelompok($data['kelompok'], 1);
$bobot = gejala::detailiterasifakta($par['id_fakta'],
1);
if ($i == 1) {
$cf = $bobot['bobot'] * $data['nilai_cf'];
} else {
$cf2 = $bobot['bobot'] * $data['nilai_cf'];
if ($cf >= 0 && $cf2 >= 0) {
$cf = $cf + $cf2 * (1 - $cf);
} elseif ($cf < 0 && $cf2 < 0) {
$cf = $cf + $cf2 * (1 + $cf);
} elseif ($cf < 0 || $cf2 < 0) {
$cf = ($cf + $cf2) / (1 - (min($cf, $cf2)));
}
}
```

**Gambar 5.8 Kode Program Perhitungan CF Kombinasi**

Penjelasan Gambar 5.8 tentang implementasi perhitungan Cf kombinasi adalah:

Baris 1 s.d 11 : Proses perhitungan cf gejala

Baris 12 s.d 19: Proses perhitungan cf kombinasi dengan pengecekan terlebih dahulu nilai cf pakar hasil optimasi dan nilai cf user



## 5.2 Implementasi Antarmuka

Implementasi antarmuka merupakan hasil perancangan antarmuka yang sebelumnya dibuat. Antarmuka sistem terdiri dari 9 antarmuka yaitu antarmuka login, antarmuka user diantaranya home, budidaya, hasil optimasi pso, hasil percobaan, dan kuis. Sedangkan pada antarmuka admin yaitu antarmuka home, parameter admin dan perhitungan cf. Implementasi antarmuka login dapat dilihat pada gambar 5.9.



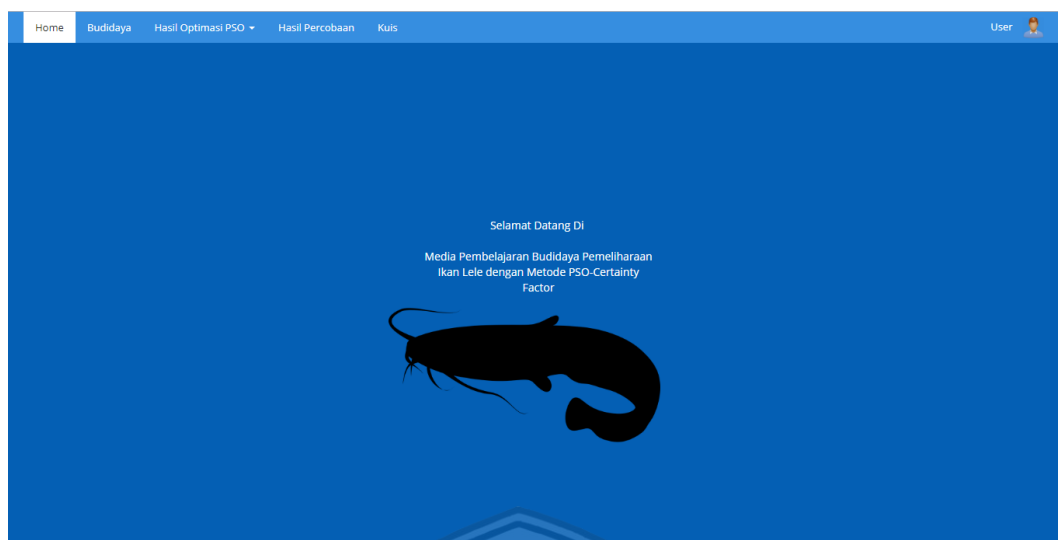
**Gambar 5.9 Implementasi Antarmuka Login**

Gambar 5.9 merupakan halaman pertama ketika sistem dijalankan yaitu halaman login.

### 5.2.1 Implementasi Antarmuka User

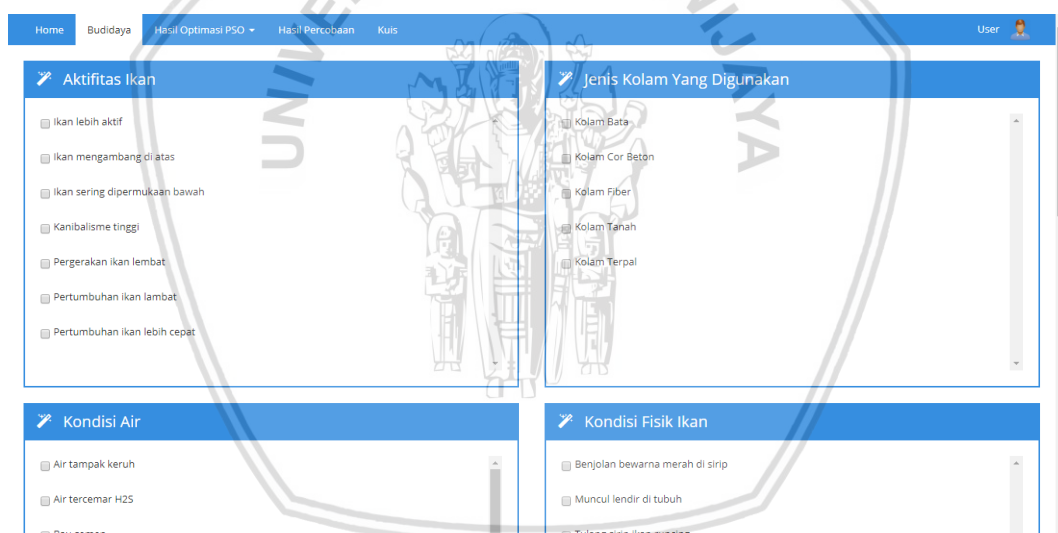
Implementasi antarmuka user merupakan halaman utama untuk pengguna yang telah login dengan level *user* pada sistem. Halaman ini berfungsi untuk menerima masukan dari pengguna. Halaman ini juga berfungsi untuk menampilkan hasil dari kombinasi fakta yang diinputkan oleh pengguna. Implementasi antarmuka user home ditunjukkan pada Gambar 5.10 sebagai berikut:





**Gambar 5.10 Implementasi Antarmuka User Home**

Gambar 5.10 merupakan tampilan awal ketika pengguna sistem telah login dengan level user dihadapkan dengan 6 menu diantaranya Home, Budidaya, Hasil Optimasi PSO, Hasil Percobaan, Kuis, dan User.



**Gambar 5.11 Implementasi Antarmuka Budidaya**

Gambar 5.11 merupakan halaman menu budidaya yang menjelaskan pengguna dihadapkan dengan 40 fakta budidaya ikan lele. 40 fakta ini digunakan untuk percobaan atau kombinasi fakta yang nantinya dapat diprediksi oleh system. Keluaran dari percobaan dan kombinasi berupa indikasi yang akan dialami oleh ikan. Pada menu Hasil Optimasi PSO terdapat semua proses pada perhitungan algoritma PSO yang ditunjukkan pada gambar 5.12 berikut.

INISIALISASI PARTIKEL					
Indikasi	Partikel 1	Partikel 2	Partikel 3	Partikel 4	Partikel 5
Terseang Bakteri	Air tampak keruh : 0.28 Pemberian anti septik 1-2 kali/ minggu : 0.01 Kolam Cor Beton : 0.39	Pemberian anti septik 1-2 kali/ minggu : 0.25 Air tampak keruh : 0.09 Kolam Cor Beton : 0.5	Air tampak keruh : 0.03 Kolam Cor Beton : 0.48 Pemberian anti septik 1-2 kali/ minggu : 0.1	Pemberian anti septik 1-2 kali/ minggu : -0.03 Air tampak keruh : 0.4 Kolam Cor Beton : 0.53	Pemberian anti septik 1-2 kali/ minggu : 0.23 Air tampak keruh : 0.16 Kolam Cor Beton : 0.49
Terseang Parasite	Pemberian anti septik 1-2 kali/ minggu : 0.07 Air tampak keruh : 0 Kolam Cor Beton : 0.44	Kolam Cor Beton : 0.36 Pemberian anti septik 1-2 kali/ minggu : 0.1 Air tampak keruh : 0.2	Air tampak keruh : 0.1 Kolam Cor Beton : 0.31 Pemberian anti septik 1-2 kali/ minggu : 0.07	Pemberian anti septik 1-2 kali/ minggu : 0.29 Air tampak keruh : 0.29 Kolam Cor Beton : 0.25	Pemberian anti septik 1-2 kali/ minggu : 0.3 Air tampak keruh : 0.31 Kolam Cor Beton : 0.26
Ikan Sehat	Air tampak keruh : 0.22 Kolam Cor Beton : 0.32 Pemberian anti septik 1-2 kali/ minggu : 0.48	Air tampak keruh : -0.08 Kolam Cor Beton : 0.49 Pemberian anti septik 1-2 kali/ minggu : 0.45	Pemberian anti septik 1-2 kali/ minggu : 0.5 Air tampak keruh : 0.24 Kolam Cor Beton : 0.6	Pemberian anti septik 1-2 kali/ minggu : 0.5 Air tampak keruh : 0.16 Kolam Cor Beton : 0.62	Pemberian anti septik 1-2 kali/ minggu : 0.56 Air tampak keruh : 0.09 Kolam Cor Beton : 0.44
Ikan Keracunan	Pemberian anti septik 1-2 kali/ minggu : 0.22 Air tampak keruh : 0.54 Kolam Cor Beton : 0.24	Air tampak keruh : 0.33 Kolam Cor Beton : 0.3 Pemberian anti septik 1-2 kali/ minggu : 0.22	Air tampak keruh : 0.52 Kolam Cor Beton : 0.39 Pemberian anti septik 1-2 kali/ minggu : -0.01	Pemberian anti septik 1-2 kali/ minggu : 0.07 Air tampak keruh : 0.54 Kolam Cor Beton : 0.19	Pemberian anti septik 1-2 kali/ minggu : 0.1 Air tampak keruh : 0.7 Kolam Cor Beton : 0.18
Ikan Mati	Kolam Cor Beton : -0.09 Pemberian anti septik 1-2 kali/ minggu : 0 Air tampak keruh : 0	Pemberian anti septik 1-2 kali/ minggu : 0 Air tampak keruh : 0.35 Kolam Cor Beton : 0.19	Pemberian anti septik 1-2 kali/ minggu : 0 Air tampak keruh : 0.06 Kolam Cor Beton : 0.12	Pemberian anti septik 1-2 kali/ minggu : 0 Air tampak keruh : 0.17 Kolam Cor Beton : -0.07	Pemberian anti septik 1-2 kali/ minggu : 0 Air tampak keruh : 0.18 Kolam Cor Beton : 0.09
Cost	-2.503	-2.42	-2.356	-2.256	-2.247

**Gambar 5.12 Implementasi Antarmuka Hasil Optimasi PSO**

Gambar 5.12 menjelaskan pada halaman user menu Hasil Optimasi PSO. User setelah memasukan fakta maka akan dilempar ke halaman hasil optimasi PSO untuk mengetahui detail berjalanya algoritma PSO mulai dari inisialisasi partikel sampai menemukan *global best*.

Home	Budidaya	Hasil Optimasi PSO	Hasil Percobaan	Kuis	User
------	----------	--------------------	-----------------	------	------

**Input CF User**

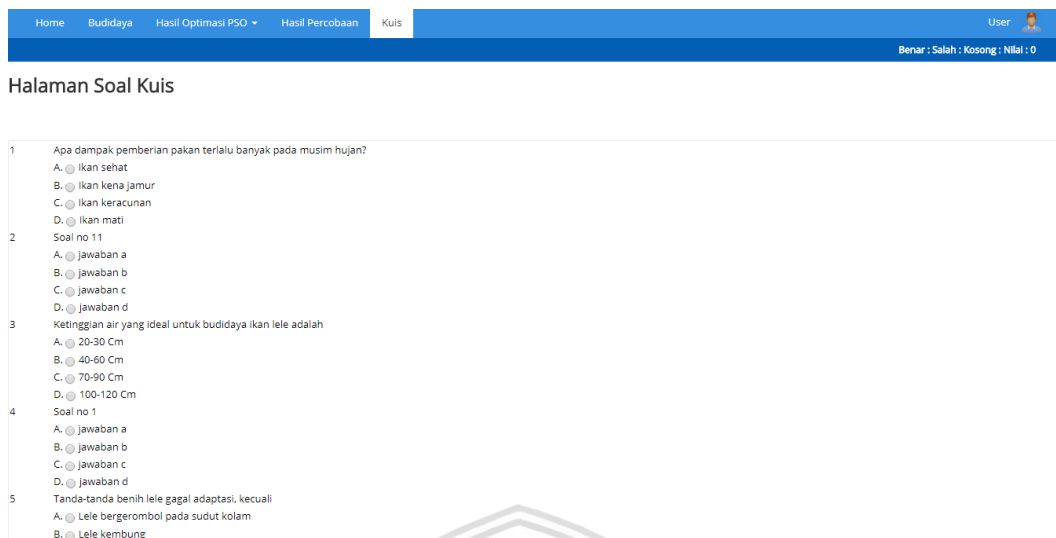
Kolam Cor Beton

Hasil Perhitungan CF					
Indikasi	Gejala	CF User	CF Pakar	CF Gejala	Hasil
P1	Kolam Cor Beton		0.42	0	0
P2	Kolam Cor Beton		0.62	0	0.587
P3	Kolam Cor Beton		0.56	0	0.860
P4	Kolam Cor Beton		0.28	0	0.480
P5	Kolam Cor Beton		0.23	0	0.084

**Kesimpulan**

**Gambar 5.13 Implementasi Antarmuka Hasil Percobaan**

Gambar 5.13 menjelaskan tampilan yang dihadapkan user ketika memilih menu hasil percobaan. Pada halaman ini user diminta untuk memasukkan nilai cf user terhadap fakta-fakta budidaya yang sebelumnya dipilih pada menu budidaya. Selanjutnya akan keluar kesimpulan akhir sistem berupa indikasi yang akan dialami ikan.



**Gambar 5.14 Implementasi Antarmuka Kuis**

Gambar 5.14 merupakan tampilan halaman jika user memilih menu kuis. Terdapat 10 soal pilihan ganda seputar pembudidayaan lele, dimana setelah menjawab soal sistem akan menampilkan skor nilai.

### 5.2.2 Implementasi Antarmuka Admin

Implementasi antarmuka admin merupakan halaman utama untuk admin. Halaman ini berfungsi untuk mengubah parameter pada proses PSO salah satunya adalah mengubah jumlah iterasi. Implementasi antarmuka admin ditunjukkan pada Gambar 5.15, Gambar 5.16, dan Gambar 5.17.



**Gambar 5.15 Implementasi Antarmuka Admin Home**

Gambar 5.15 merupakan halaman pertama yang akan ditampilkan ketika pengguna login dengan level admin. Dimana terdapat 3 menu yang ditampilkan yaitu Home, Parameter Admin, dan Perhitungan CF.

Home Parameter Admin Perhitungan CF Admin			
Parameter Admin			
#	Parameter	Nilai	#
1	random min	0	
2	random max	1	
3	c1	2	
4	c2	2	
5	wmax	0.9	
6	wmin	0.4	
7	iterasi max	10	
8	kecepatan inisial	0	
9	jumlah partikel	5	

**Gambar 5.16 Implementasi Antarmuka Parameter Admin**

Gambar 5.16 merupakan tampilan dari menu parameter admin, dimana admin dapat merubah parameter PSO seperti jumlah iterasi, dll.

Home	Parameter Admin	Certainty Factor	Admin
------	-----------------	------------------	-------

**Aktivitas Ikan**

☐ Ikan lebih aktif --pilih--

☐ Ikan mengambang di atas --pilih--

☐ Ikan sering dipermukaan bawah --pilih--

☐ Kanibalisme tinggi --pilih--

☐ Pergerakan ikan lambat --pilih--

☐ Pertumbuhan ikan lambat --pilih--

**Jenis Kolam Yang Digunakan**

☐ Kolam Bata --pilih--

☐ Kolam Cor Beton --pilih--

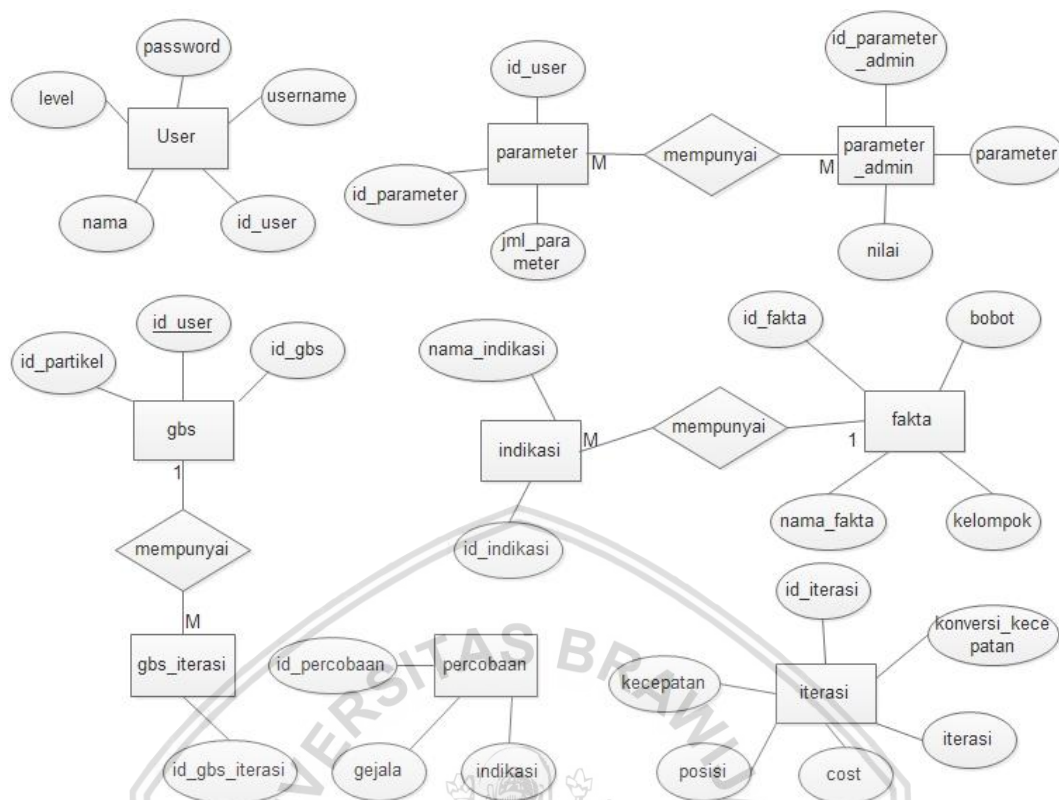
☐ Kolam Fiber --pilih--

**Gambar 5.17 Implementasi Antarmuka Certainty Factor**

Gambar 5.17 merupakan tampilan dari menu certainty factor dimana hanya bisa diakses oleh admin dan ini digunakan sebagai pengujian sistem.

### 5.3 Implementasi Database

Implementasi database merupakan hasil dari implementasi perancangan sistem. Implementasi database digunakan untuk memberikan gambaran database yang dibuat. Entity Relationship Diagram digunakan untuk mendokumentasikan data dengan mengidentifikasi entitas dan hubungannya. Implementasi database ditunjukkan pada Gambar 5.18



**Gambar 5.18 Implementasi Database**

Penejelasan Gambar 5.18 adalah sebagai berikut :

1. Entitas User : Entitas user digunakan untuk memisahkan pengguna antara admin dan pengguna biasa. Pada entitas user terdapat kolom username dan tabel untuk memisahkan antara user dan admin.
2. Entitas Gbs : Entitas Gbs digunakan untuk menyimpan hasil Global Best pada perhitungan PSO. Pada entitas ini mempunyai hubungan dengan tabel Gbs\_iterasi.
3. Entitas Gbs\_iterasi : Entitas Gbs\_iterasi digunakan untuk menyimpan hasil Global best setiap iterasi. Pada entitas ini mempunyai hubungan dengan tabel Gbs.
4. Entitas Parameter : Entitas Parameter digunakan untuk menyimpan jumlah partikel dari masukan user.
5. Entitas Parameter Admin : Entitas Parameter\_admin digunakan untuk menyimpan masukan dari admin berupa parameter PSO.
6. Entitas indikasi : Entitas indikasi digunakan untuk menyimpan semua indikasi pada sistem ini diantaranya Terserang Bakteri, Terserang Parasit, Ikan Sehat, Ikan Keracunan, dan Ikan Mati.
7. Entitas Gejala : Entitas Gejala digunakan untuk menyimpan seluruh hasil dari perhitungan PSO termasuk nilai cost, kecepatan, posisi, dan konversi kecepatan.

8. Entitas Iterasi : Entitas Iterasi digunakan untuk menyimpan seluruh hasil perhitungan PSO setiap iterasinya.
9. Entitas Fakta : Entitas fakta digunakan untuk menyimpan fakta-fakta budaya sebanyak 40 fakta dikombinasikan dengan 5 indikasi.





## BAB 6 PENGUJIAN

Bab ini menjelaskan hasil pengujian dan pembahasan dari implementasi metode *Particle Swarm Optimization – Certainty Factor* untuk mengenali kondisi ikan lele. Pengujian yang dilakukan meliputi pengujian parameter PSO, dan pengujian akurasi sistem.

### 6.1 Pengujian Parameter PSO

Pengujian parameter PSO bertujuan untuk mengetahui pengaruh parameter PSO dalam menghasilkan nilai *cost* terbaik. Parameter yang diuji coba adalah kombinasi bobot inersia, kombinasi koefisien akselerasi, jumlah iterasi, dan ukuran swarm.

Fakta yang dipakai selama pengujian adalah :

1. Kolam Tanah
2. Air tampak keruh
3. Ph air berubah-ubah
4. Ukuran benih dibawah 2 cm
5. Keseragaman bibit diatas 80%

#### 6.1.1 Pengujian Bobot Inersia

Pengujian bobot inersia bertujuan untuk mengetahui kombinasi bobot inersia maksimal ( $W_{max}$ ) dan bobot inersia minimal ( $W_{min}$ ) yang tepat guna memperoleh nilai *cost* yang optimal. Nilai bobot inersia maksimal yang digunakan adalah 0.9, 0.8, dan 0.7 sedangkan nilai bobot inersia minimal yang digunakan adalah 0.2, 0.3, 0.4. kombinasi bobot inersia maksimal dan minimal dievaluasi berdasarkan rata rata nilai *cost* terbaik. Untuk memperoleh rata-rata nilai secara keseluruhan, dilakukan percobaan sebanyak 10 kali untuk tiap kombinasi bobot inersia. Hasil pengujian kombinasi bobot inersia ditunjukkan pada Tabel 6.1 sampai dengan Tabel 6.3.

**Tabel 6.1 Hasil Pengujian Bobot Inersia Maksimal 0,9**

Uji coba ke-	Nilai Cost Terbaik		
	$\omega_{min} = 0,2$	$\omega_{min} = 0,3$	$\omega_{min} = 0,4$
1	-2.807	-3.012	-3.005
2	-3.284	-2.859	-3.23
3	-2.993	-2.974	-3.007
4	-2.829	-3.084	-3.069
5	-2.705	-3.123	-2.89
6	-3.204	-3.219	-3.189
7	-3.032	-3.055	-3.05

8	-2.901	-2.98	-3.178
9	-2.856	-3.066	-3.285
10	-2.902	-3.196	-3.114
Rata-rata	-2.951	-3.065	-3.121

Tabel 6.2 Hasil Pengujian Bobot Inersia Maksimal 0,8

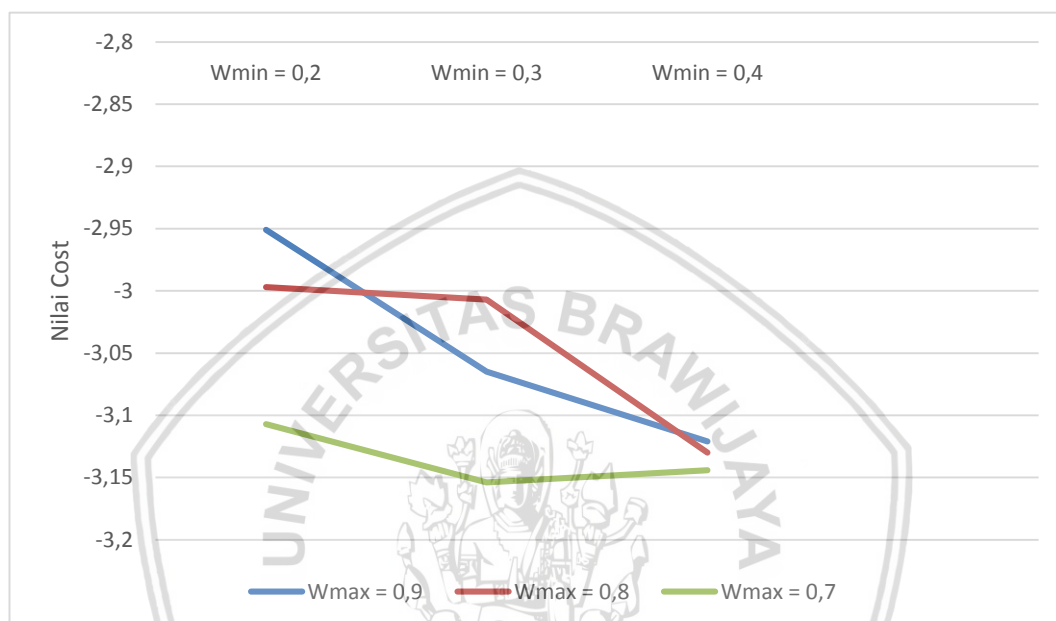
Uji coba ke-	Nilai Cost Terbaik		
	$\omega_{min} = 0,2$	$\omega_{min} = 0,3$	$\omega_{min} = 0,4$
1	-3.113	-3.037	-2.974
2	-3.009	-3.061	-3.209
3	-3.065	-3.137	-3.034
4	-2.925	-3.018	-3.001
5	-3.12	-3.068	-3.108
6	-3.037	-3.166	-3.191
7	-2.914	-3.16	-3.228
8	-3.26	-3.183	-3.039
9	-3.12	-3.048	-3.194
10	-2.918	-2.978	-3.322
Rata-rata	-2.997	-3.007	-3.130

Tabel 6.3 Hasil Pengujian Bobot Inersia Maksimal 0,7

Uji coba ke-	Nilai Cost Terbaik		
	$\omega_{min} = 0,2$	$\omega_{min} = 0,3$	$\omega_{min} = 0,4$
1	-3.149	-2.998	-3.207
2	-2.973	-3.288	-3.106
3	-3.263	-3.084	-3.155
4	-3.246	-3.096	-3.148
5	-3.219	-3.259	-3.023
6	-3.178	-3.264	-3.251
7	-3.191	-3.1	-3.118
8	-2.904	-3.297	-3.323

9	-3.019	-2.946	-3.195
10	-2.929	-3.142	-2.918
Rata-rata	-3.107	-3.154	-3.144

Grafik hasil pengujian kombinasi bobot inersia maksimum dan bobot inersia minimal ditunjukkan pada Gambar 6.1.



**Gambar 6.1 Grafik Pengujian Bobot Inersia**

Bobot inersia merupakan mekanisme untuk mengontrol daya eksplorasi dan eksploitasi partikel. Nilai bobot inersia yang besar meningkatkan daya eksplorasi sehingga keragaman partikel dalam swarm meningkat sedangkan nilai bobot inersia yang kecil meningkatkan daya eksploitasi namun rentan terhadap hilangnya daya eksplorasi partikel (Engelbrecht, 2007). Eksplorasi cenderung memperluas ruang pencarian secara global sebaliknya eksploitasi cenderung berfokus pada pencarian solusi pada optimum lokal. Oleh karena keseimbangan antara kedua hal tersebut sangat menentukan PSO dalam menemukan solusi optimum, maka pemilihan inisialisasi nilai bobot inersia yang tepat akan berbeda, bergantung pada karakteristik permasalahan yang dihadapi (Shi, et al 1998).

Berdasarkan grafik hasil pengujian bobot inersia pada Gambar 6.1 maka diperoleh kombinasi bobot inersia terbaik dengan menerapkan bobot inersia maksimal 0,9 dan bobot inersia minimal 0,2. Hal ini dapat dilihat pada grafik pengujian bobot inersia bahwa pada saat bobot inersia maksimal 0,9 dan bobot inersia minimal 0,2 nilai cost paling rendah yang berarti paling mendekati 0 (nol) dijadikan sebagai solusi terbaik.

### 6.1.2 Pengujian Koefisien Akselerasi

Pengujian koefisien akselerasi dilakukan untuk mengetahui kombinasi koefisien akselerasi 1 dan koefisien akselerasi 2 terbaik guna memperoleh nilai *cost* terbaik yang mendekati 0 (nol). Karena PSO akan menghasilkan hasil yang berbeda setiap dijalankan maka dibutuhkan percobaan 10 kali untuk setiap kombinasi koefisien akselerasi. Hasil pengujian koefisien akselerasi ditunjukkan pada Tabel 6.4 sampai Tabel 6.

**Tabel 6.4 Hasil Percobaan Koefisien Akselerasi1 = 1**

Uji coba ke-	Nilai Cost Terbaik		
	$C_2 = 1$	$C_2 = 1,5$	$C_2 = 2$
1	-3.038	-2.77	-3.005
2	-3.069	-3.23	-2.89
3	-3.152	-3.085	-2.937
4	-2.964	-2.89	-3.069
5	-3.264	-2.93	-3.01
6	-3.122	-3.08	-2.951
7	-3.187	-3.17	-2.87
8	-3.03	-3.017	-3.178
9	-2.993	-3.158	-2.891
10	-3.163	-2.92	-2.916
Rata-rata	-3.106	-3.087	-2.992

**Tabel 6.5 Hasil Percobaan Koefisien Akselerasi1 = 1,5**

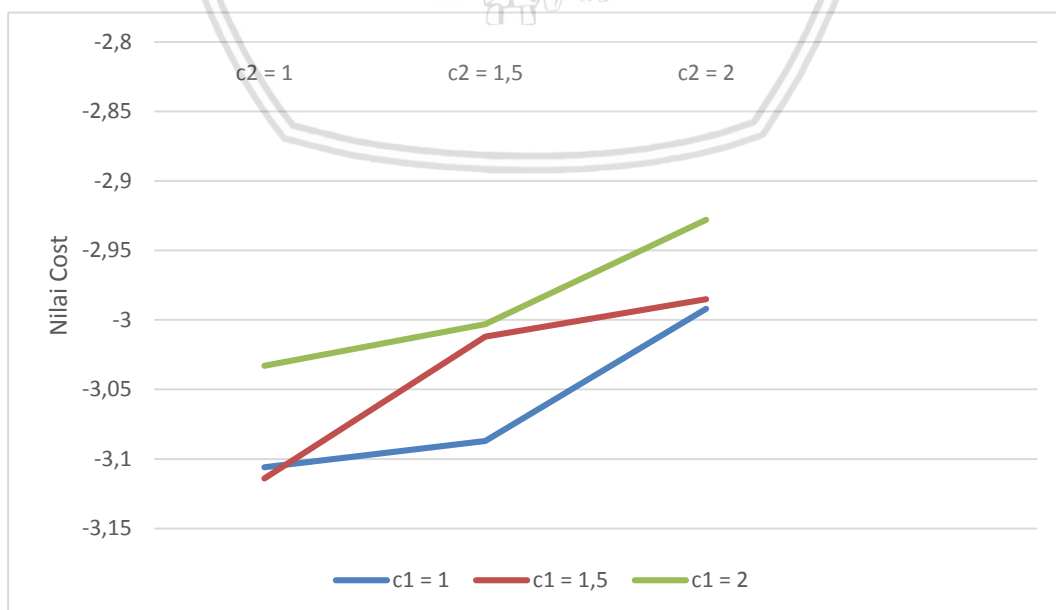
Uji coba ke-	Nilai Cost Terbaik		
	$C_2 = 1$	$C_2 = 1,5$	$C_2 = 2$
1	-3.133	-3.054	-2.862
2	-3.183	-3.136	-3.053
3	-3.24	-3.192	-2.762
4	-3.101	-2.915	-2.88
5	-3.19	-3.017	-2.95
6	-3.142	-2.95	-2.98
7	-3.108	-3.087	-3.012
8	-3.120	-3.164	-3.178

9	-3.148	-2.893	-3.044
10	-2.973	-3.012	-2.86
Rata-rata	-3.114	-3.012	-2.985

**Tabel 6.5 Hasil Percobaan Koefisien Akselerasi1 = 2**

Uji coba ke-	Nilai Cost Terbaik		
	$C_2 = 1$	$C_2 = 1,5$	$C_2 = 2$
1	-3.021	-3.178	-2.936
2	-2.891	-3.264	-3.011
3	-3.221	-2.851	-2.782
4	-3.038	-3.036	-2.943
5	-3.087	-2.875	-2.918
6	-3.018	-2.907	-3.008
7	-3.083	-3.053	-2.852
8	-2.915	-2.922	-2.907
9	-3.083	-2.863	-3.039
10	-2.973	-3.084	-2.881
Rata-rata	-3.033	-3.003	-2.928

Grafik hasil pengujian kombinasi koefisien akselerasi 1 dan koefisien akselerasi 2 ditunjukkan pada Gambar 6.2



**Gambar 6.2 Grafik Pengujian Koefisien Akselerasi**

Koefisien akselerasi bertugas mengontrol pergerakan partikel pada ruang pencarian. Secara umum, nilai koefisien akselerasi adalah sama dan bersifat statis. Nilai koefisien akselerasi 1 dan akselerasi 2 yang besar menyebabkan pergerakan partikel menempati posisi baru yang relative lebih jauh sehingga kemampuan eksplorasi partikel menjadi lebih baik namun rentan menyimpang dari batas ruang pencarian. Sebaliknya, nilai akselerasi 1 dan akselerasi 2 yang kecil menyebabkan pergerakan partikel menjadi terbatas sehingga memungkinkan terperangkap pada pencarian optimum lokal. Kondisi lain adalah ketika nilai koefisien akselerasi 1 lebih besar dibandingkan akselerasi 2 maka pergerakan partikel cenderung mengarah pada posisi terbaik dari setiap partikel itu sendiri atau Personal best dan sebaliknya jika nilai akselerasi 2 lebih besar dibandingkan dengan akselerasi 1 maka pergerakan partikel cenderung mengarah ke posisi terbaik partikel secara keseluruhan atau Global Best (Ahmed, et al., 2008). Sehingga pemilihan koefisien bisa bervariasi tergantung pada karakteristik permasalahan (Valle, et al., 2008).

Berdasarkan grafik pengujian koefisien akselerasi pada Gambar 6.2 diperoleh kombinasi koefisien akselerasi terbaik adalah dengan menerapkan kombinasi akselerasi 1 bernilai 2 dan akselerasi 2 bernilai 2. Karakteristik permasalahan berdasarkan hasil pengujian menunjukkan bahwa perubahan nilai cost cenderung dipengaruhi oleh koefisien akselerasi 2 yaitu semakin besar koefisien akselerasi 2 maka semakin mendekati 0 (nol) nilai cost yang dihasilkan. Hal tersebut juga berlaku untuk koefisien akselerasi 2 yaitu semakin besar nilai koefisien akselerasi maka nilai cost yang dihasilkan semakin mendekati 0 (nol). Sehingga dapat disimpulkan bahwa pada permasalahan ini, kemampuan pencarian solusi optimum oleh partikel-partikel PSO dipengaruhi oleh posisi terbaik partikel itu sendiri atau Personal Best dan posisi terbaik partikel secara keseluruhan atau Global Best.

### 6.1.3 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi dilakukan untuk mengetahui jumlah iterasi yang tepat untuk memperoleh nilai cost terbaik yaitu mendekati nilai 0 (nol). Jumlah iterasi yang digunakan adalah kelipatan 10. PSO merupakan algoritma stokastik sehingga akan menghasilkan hasil yang berbeda setiap kali program dijalankan (Mahmudy, 2015), karenanya untuk memperoleh rata-rata nilai keseluruhan, dilakukan percobaan sebanyak 10 kali setiap jumlah iterasi. Hasil pengujian jumlah iterasi ditunjukkan pada Tabel 6.6.

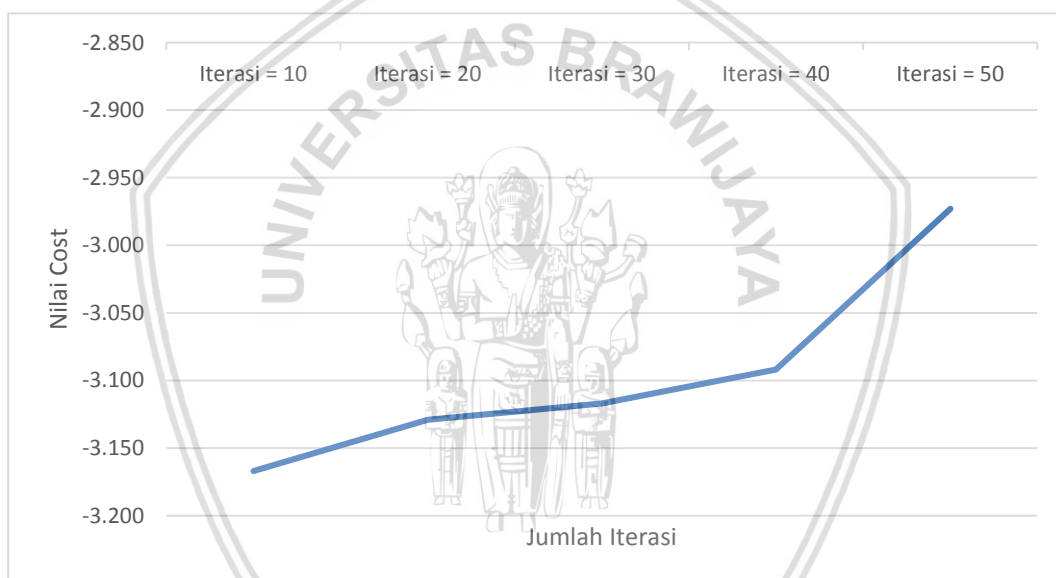
**Tabel 6.6 Hasil Pengujian Jumlah Iterasi**

Uji coba ke-	Nilai Cost Terbaik				
	Iterasi = 10	Iterasi = 20	Iterasi = 30	Iterasi = 40	Iterasi = 50
1	-3.262	-3.162	-3.166	-3.031	-3.121
2	-3.084	-3.085	-3.108	-2.862	-3.018



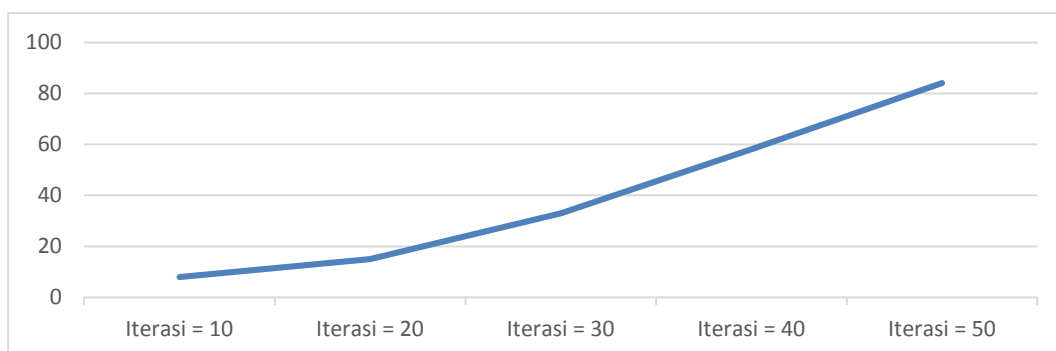
3	-3.317	-3.261	-3.116	-3.045	-2.84
4	-3.195	-3.196	-3.092	-3.671	-2.911
5	-3.076	-3.127	-3.071	-3.018	-3.015
6	-3.214	-3.089	-3.221	-3.077	-3.22
7	-3.088	-3.141	-2.952	-3.27	-2.811
8	-3.255	-3.139	-3.099	-2.961	-2.906
9	-3.061	-3.021	-3.204	-3.019	-2.921
10	-3.118	-3.072	-3.139	-3.145	-3.079
Rata-rata	-3.167	-3.129	-3.117	-3.092	-2.973

Grafik pengujian jumlah iterasi ditunjukkan pada Gambar 6.3



Gambar 6.3 Grafik Pengujian Jumlah Iterasi

Grafik hasil pengujian jumlah iterasi terhadap waktu komputasi ditunjukkan pada Gambar 6.4



Gambar 6.4 Grafik Waktu Komputasi Berdasarkan Iterasi

Berdasarkan grafik hasil pengujian jumlah iterasi pada Gambar 6.3 diperoleh jumlah iterasi terbaik dengan rata-rata nilai cost terkecil adalah 50 iterasi. Jumlah iterasi yang dibutuhkan untuk memperoleh solusi terbaik bergantung pada jenis permasalahan yang dihadapi. Jumlah iterasi yang terlalu sedikit menyebabkan pencarian solusi penyelesaian menjadi prematur atau berhenti sebelum solusi optimum didapatkan. Sebaliknya, jumlah iterasi yang terlalu banyak menyebabkan kompleksitas komputasi semakin meningkat walaupun sebenarnya langkah tersebut tidak diperlukan (Engelbrecht, 2007). Hal ini ditunjukkan pada Gambar 6.4 dimana semakin besar jumlah iterasi maka waktu yang dibutuhkan untuk menyelesaikan komputasi juga semakin lama. Oleh karena itu, direkomendasikan menggunakan jumlah iterasi sebanyak 20 iterasi karena mulai iterasi ke-30, perubahan nilai cost tidak terlalu signifikan dan waktu komputasi yang dibutuhkan lebih efisien. Hal tersebut disebabkan partikel melakukan pencarian pada wilayah yang sama namun tidak menemukan solusi yang signifikan lebih baik dari solusi yang diperoleh pada iterasi sebelumnya sehingga nilai cost yang dihasilkan relative sama.

#### 6.1.4 Pengujian Ukuran Swarm

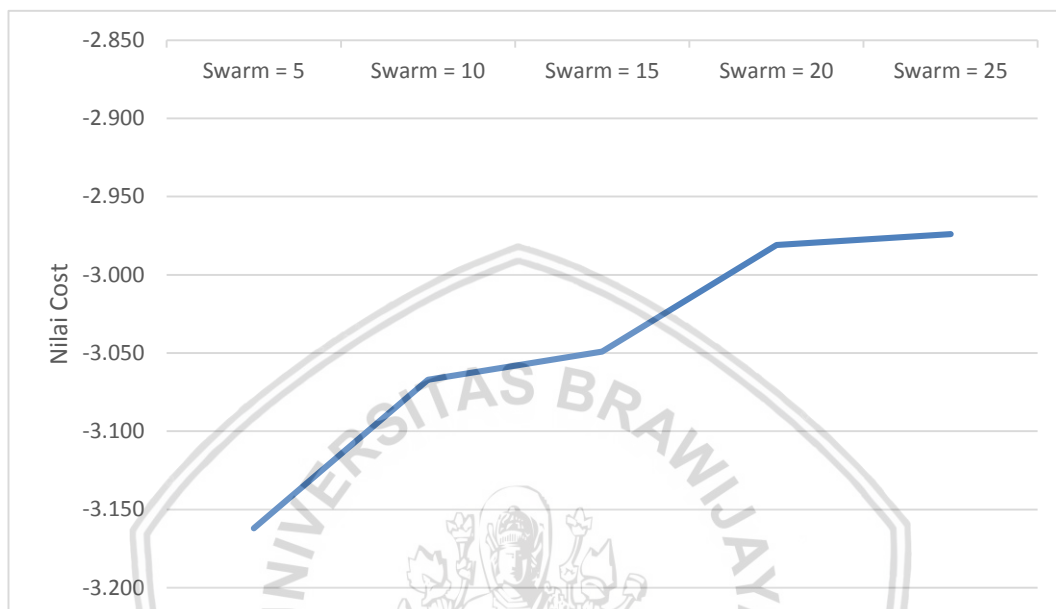
Pengujian swarm dilakukan untuk mengetahui jumlah populasi yang dibutuhkan untuk memperoleh nilai cost terbaik. Jumlah partikel dalam swarm yang digunakan adalah kelipatan 5. Ukuran swarm optimum dievaluasi berdasarkan nilai rata-rata cost terbaik. PSO merupakan algoritma stokastik sehingga akan menghasilkan hasil berbeda setiap kali program dijalankan (Mahmudy, 2015) karenanya dalam percobaan dilakukan sebanyak 10 kali untuk setiap ukuran swarm. Hasil pengujian ukuran swarm ditunjukkan pada Tabel 6.7

**Tabel 6.7 Hasil Pengujian Ukuran Swarm**

Uji coba ke-	Nilai Cost Terbaik				
	Partikel = 5	Iterasi = 10	Iterasi = 15	Iterasi = 20	Iterasi = 25
1	-3.231	-3.072	-3.013	-3.047	-2.859
2	-3.28	-3.094	-3.164	-2.953	-3.074
3	-3.193	-3.129	-3.082	-3.08	-2.95
4	-3.218	-3.031	-3.101	-3.029	-3.031
5	-3.165	-2.972	-3.081	-3.153	-3.104
6	-3.018	-3.005	-2.941	-3.005	-2.821
7	-3.061	-3.108	-2.903	-2.641	-2.948
8	-3.275	-3.254	-3.116	-3.041	-3.085
9	-3.093	-3.018	-3.033	-3.121	-3.029
10	-3.201	-3.18	-3.053	-2.842	-2.818

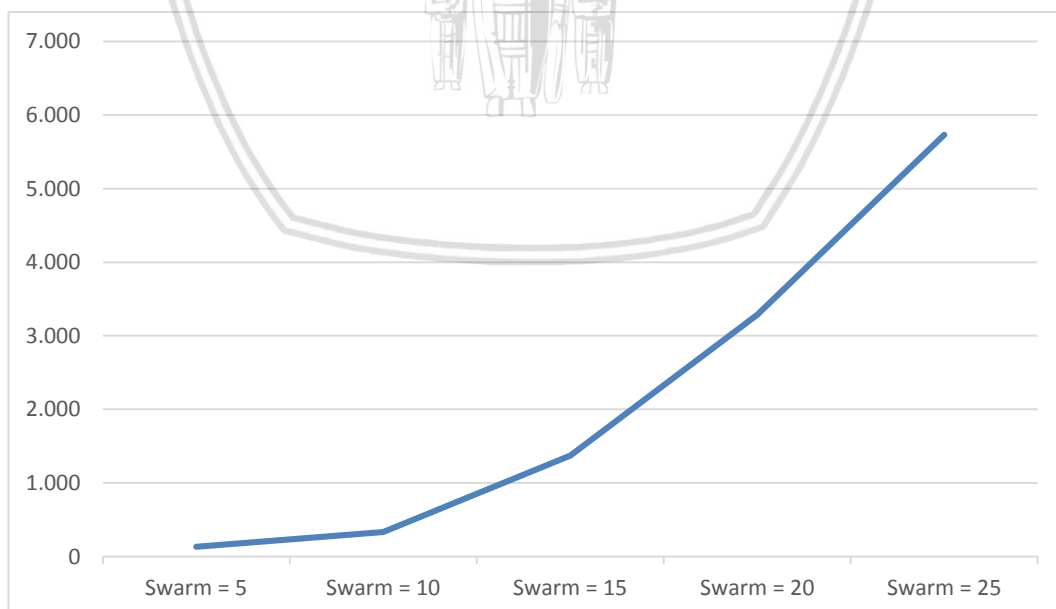
Rata-rata	-3.162	-3.076	-3.049	-2.981	-2.974
-----------	--------	--------	--------	--------	--------

Grafik hasil pengujian ukuran swarm ditunjukkan pada Gambar 6.5.



**Gambar 6.5 Grafik Pengujian Ukuran Swarm**

Grafik pengujian ukuran swarm terhadap waktu komputasi ditunjukkan pada Gambar 6.6



**Gambar 6.6 Grafik Pengujian Ukuran Swarm Terhadap Waktu Komputasi**

Berdasarkan hasil pengujian ukuran swam pada Gambar 6.5 diperoleh jumlah partikel terbaik dengan rata-rata nilai *cost* terkecil adalah 25 partikel. Semakin

besar jumlah partikel maka semakin tinggi tingkat keragaman partikel dalam swarm sehingga menyebabkan skema inisialisasi partikel juga semakin baik. Ukuran swarm yang besar memungkinkan partikel PSO untuk mengeksplorasi ruang pencarian semakin luas sehingga peluang untuk mendapatkan solusi penyelesaian optimum juga semakin besar. Namun demikian, semakin besar jumlah partikel menyebabkan waktu komputasi yang dibutuhkan juga semakin kompleks (Engelbrecht, 2007). Hal tersebut ditunjukkan pada Gambar 6.6 dimana semakin besar ukuran swarm maka waktu yang dibutuhkan untuk menyelesaikan komputasi juga semakin lama. Oleh karena itu, direkomendasikan menggunakan ukuran swarm sebanyak 10 partikel karena dengan jumlah partikel sebanyak 10, partikel-partikel PSO sudah mampu menghasilkan nilai cost yang cukup rendah daripada sebelumnya dan dengan waktu komputasi yang lebih efisien.

### 6.1.5 Pengujian Akurasi

Pengujian akurasi adalah pengujian yang dilakukan untuk mengetahui tingkat performa kecocokan keluaran sistem dengan keluaran dari pakar. Pada pengujian akurasi ini juga bertujuan untuk mengetahui perbandingan metode sebelum hybrid PSO-Certainty *Caftor* dan sesudah dilakukan hybrid sehingga dapat diketahui peningkatan performa sistem. Pada pengujian akurasi dilakukan 20 data uji yang diperoleh dari wawancara dan menghasilkan data kombinasi budidaya ikan lele. Dari data 30 data uji dilakukan analisa kesesuaian antara hasil sistem dengan hasil yang dilakukan pakar. Pakar menetapkan 20 kasus beserta hasil indikasi yang nantinya dievaluasi dengan hasil keluaran sistem menggunakan metode PSO-Certainty *Factor*. Hasil pengujian akurasi sesudah hybrid dan sebelum hybrid PSO-Certainty *Factor* dari 20 data kasus yang telah diuji ditunjukkan pada Tabel 6.8 dan Tabel 6.9.

**Tabel 6.8 Hasil Pengujian Akurasi Hybrid PSO-Certainty Factor dengan Hasil Pakar**

Percobaan	Fakta yang dipilih	Hasil Indikasi Sistem	Hasil Indikasi Pakar	Akurasi Sistem
1	6.1.5.1.1.1.1 Kolam cor beton (1) 6.1.5.1.1.1.2 Air tampak keruh (0,4) 6.1.5.1.1.1.3 Pemberian antiseptik 1-2kali /minggu (1)	Ikan Sehat	Ikan Sehat	1
2	1. Kolam cor beton (1) 2. Bau semen (1)	Ikan Keracunan	Ikan Keracunan	1

	3. Warna ikan pucat (0,4) 4. Keseragaman bibit dibawah 80% (1)			
3	Kolam tanah (1) Ikan mengambang (1) Air tampak keruh (1) Pemberian antiseptik lebih dari 2kali/minggu (1)	Ikan Sehat	Ikan Sehat	1
4	1. Kolam terpal (1) 2. Temperatur suhu tidak stabil (0,4) 3. Pemberian antiseptik 1-2kali/minggu (1) 4. Keseragaman bibit diatas 80% (1)	Ikan Sehat	Ikan Sehat	1
5	1. Kolam terpal (1) 2. Mencul lendir ditubuh (0,4) 3. Pergerakan ikan lambat (0,4)	Ikan Terserang Parasit	Ikan Terserang Bakteri	0
6	1. Kolam cor beton (1) 2. Ikan mengambang (1) 3. Benjolan warna merah disirip (1)	Ikan Terserang Bakteri	Ikan Terserang Bakteri	1
7	1. Kolam fiber (1) 2. Suhu diatas 30 derajat (0,4) 3. Pertumbuhan ikan lambat (1) 4. Keseragaman bibit diatas 80% (1)	Ikan Sehat	Ikan Sehat	1
8	1. Kolam bata (1) 2. Ikan sering dipermukaan bawah (1) 3. Suhu diatas 30 derajat	Ikan Terserang Parasit	Ikan Terserang Parasit	1

	(0,4)			
9	1. Kolam bata (1) 2. Kanibalisme tinggi (0,4) 3. Keseragaman bibit dibawah 80% (1)	Ikan Mati	Ikan Mati	1
10	1. Kolam cor beton (1) 2. Pemberian prebiotik 20-40ppm/minggu (1) 3. Pemberian antiseptik 1-2 kali/minggu (1) 4. Suhu tidak stabil (0,4)	Ikan Sehat	Ikan Sehat	1
11	1. Kolam fiber (1) 2. Pembuangan kotoran 2-3kali/minggu (1) 3. Pemberian antiseptik 1-2kali/minggu (1) 4. Ikan lebih aktif (1)	Ikan Sehat	Ikan Sehat	1
12	1. Kolam terpal (1) 2. Muncul lendir ditubuh (1) 3. Keseragaman bibit ikan ideal (0,4) 4. Pemberian antiseptik 1-2kali/minggu (1) 5. Ikan lebih aktif (0,4) 6. Penebaran fermentasi diatas 500ml (1)	Ikan Terseparang Parasit	Ikan Terseparang Parasit	1
13	1. Kolam bata (1) 2. Pergerakan ikan lambat (1) 3. Air tampak keruh (1) 4. Temperatur suhu tidak stabil (0)	Ikan Terseparang Bakteri	Ikan Terseparang Bakteri	1
14	1. Ukuran benih dibawah 2 cm (1) 2. Pemberian antiseptik lebih dari 2kali/minggu	Ikan Mati	Ikan Mati	1



	(1)			
15	1. Kolam tanah (1) 2. Pertumbuhan ikan lebih cepat (0,4) 3. Terdapat bahan organik dikolam (0,4) 4. Ukuran benih lebih dari 2cm (1) 5. Pemberian antiseptik lebih dari 2kali/minggu 6. PH air berubah-ubah (0) 7. Air tampak keruh (1)	Ikan Sehat	Ikan Sehat	1
16	1. Kolam terpal (1) 2. Pertumbuhan ikan lambat (0,4) 3. Benjolan warna merah disirip (1) 4. Keseragaman bibit dibawah 80% (1)	Ikan Terserang Bakteri	Ikan Terserang Bakteri	1
17	1. Kolam fiber (1) 2. Ikan mengambang (1) 3. Air tercemar H <sub>2</sub> S (0,4) 4. PH air berubah-ubah(0) 5. Ukuran kepala terlalu besar (1)	Ikan Terserang Parasit	Ikan Terserag Parasit	1
18	1. Terdapat bahan organik dikolam (0,4) 2. Pertumbuhan ikan cepat (1)	Ikan Sehat	Ikan Sehat	1
19	1. Ukuran benih kurang dari 2 cm (1)	Ikan Mati	Ikan Terserang Parasit, Ikan Mati	0
20	1. Kolam fiber (1)	Ikan Sehat	Ikan Sehat	1

	2. Air tercemar H <sub>2</sub> S (0,4) 3. Pemberian antiseptik lebih dari 2kali/minggu (1) 4. Pembuangan kotoran 2-3kali/minggu (1) 5. Ukuran benih dibawah 2cm (1)			
--	--	--	--	--

Hasil Akurasi bernilai 1 berarti keluaran dari perhitungan sistem sama dengan hasil dari pakar, sebaliknya jika hasil akurasi 0 artinya keluaran sistem tidak sama dengan pakar. Berdasarkan Tabel 6.8 dilakukan perhitungan akurasi menggunakan Persamaan 2.11 dan menghasilkan nilai akurasi sebagai berikut.

$$\text{Nilai akurasi} = \frac{18}{20} \times 100 \% = 90\%$$

**Tabel 6.9 Hasil Pengujian Akurasi Certainty Factor dengan Hasil Pakar**

Percobaan	Fakta yang dipilih	Hasil Indikasi Sistem	Hasil Indikasi Pakar	Akurasi Sistem
1	1. Kolam cor beton (1) 6.1.5.1.1.1.4 Air tampak keruh (0,4) 6.1.5.1.1.1.5 Pemberian antiseptik 1-2kali /minggu (1)	Ikan Sehat	Ikan Sehat	1
2	1. Kolam cor beton (1) 2. Bau semen (1) 3. Warna ikan pucat (0,4) 4. Keseragaman bibit dibawah 80% (1)	Ikan Keracunan	Ikan Keracunan	1
3	1. Kolam tanah (1) 2. Ikan mengambang (1) 3. Air tampak keruh (1) 4. Pemberian antiseptik lebih dari 2kali/minggu (1)	Ikan Sehat	Ikan Sehat	1

4	1. Kolam terpal (1) 2. Temperatur suhu tidak stabil (0,4) 3. Pemberian antiseptik 1-2kali/minggu (1) 4. Keseragaman bibit diatas 80% (1)	Ikan Sehat	Ikan Sehat	1
5	1. Kolam terpal (1) 2. Mencul lendir ditubuh (0,4) 3. Pergerakan ikan lambat (0,4)	Ikan Terserang Parasit	Ikan Terserang Bakteri	0
6	1. Kolam cor beton (1) 2. Ikan mengambang (1) 3. Benjolan warna merah disirip (1)	Ikan Terserang Bakteri	Ikan Terserang Bakteri	1
7	1. Kolam fiber (1) 2. Suhu diatas 30 derajat (0,4) 3. Pertumbuhan ikan lambat (1) 4. Keseragaman bibit diatas 80% (1)	Ikan Sehat	Ikan Sehat	1
8	1. Kolam bata (1) 2. Ikan sering dipermukaan bawah (1) 3. Suhu diatas 30 derajat (0,4)	Ikan Terserang Parasit	Ikan Terserang Parasit	1
9	1. Kolam bata (1) 2. Kanibalisme tinggi (0,4) 3. Keseragaman bibit dibawah 80% (1)	Ikan Mati	Ikan Mati	1
10	1. Kolam cor beton (1) 2. Pemberian prebiotik 20-40ppm/minggu (1)	Ikan Sehat	Ikan Sehat	1

	3. Pemberian antiseptik 1-2 kali/minggu (1) 4. Suhu tidak stabil (0,4)			
11	1. Kolam fiber (1) 2. Pembuangan kotoran 2-3 kali/minggu (1) 3. Pemberian antiseptik 1-2 kali/minggu (1) 4. Ikan lebih aktif (1)	Ikan Sehat	Ikan Sehat	1
12	1. Kolam terpal (1) 2. Muncul lendir ditubuh (1) 3. Keseragaman bibit ikan ideal (0,4) 4. Pemberian antiseptik 1-2 kali/minggu (1) 5. Ikan lebih aktif (0,4) 6. Penebaran fermentasi diatas 500ml (1)	Ikan Sehat	Ikan Terserang Parasit	0
13	1. Kolam bata (1) 2. Pergerakan ikan lambat (1) 3. Air tampak keruh (1) 4. Temperatur suhu tidak stabil (0)	Ikan Terserang Bakteri	Ikan Terserang Bakteri	1
14	1. Ukuran benih dibawah 2 cm (1) 2. Pemberian antiseptik lebih dari 2 kali/minggu (1)	Ikan Mati	Ikan Mati	1
15	1. Kolam tanah (1) 2. Pertumbuhan ikan lebih cepat (0,4) 3. Terdapat bahan organik dikolam (0,4) 4. Ukuran benih lebih dari	Ikan Sehat	Ikan Sehat	1

	<p>2cm (1)</p> <p>5. Pemberian antiseptik lebih dari 2kali/minggu</p> <p>6. PH air berubah-ubah (0)</p> <p>7. Air tampak keruh (1)</p>			
16	<p>1. Kolam terpal (1)</p> <p>2. Pertumbuhan ikan lambat (0,4)</p> <p>3. Benjolan warna merah disirip (1)</p> <p>4. Keseragaman bibit dibawah 80% (1)</p>	Ikan Terserang Bakteri	Ikan Terserang Bakteri	1
17	<p>1. Kolam fiber (1)</p> <p>2. Ikan mengambang (1)</p> <p>3. Air tercemar H<sub>2</sub>S (0,4)</p> <p>4. PH air berubah-ubah(0)</p> <p>5. Ukuran kepala terlalu besar (1)</p>	Sehat	Ikan Terserag Parasit	0
18	<p>1. Terdapat bahan organik dikolam (0,4)</p> <p>2. Pertumbuhan ikan cepat (1)</p>	Ikan Sehat	Ikan Sehat	1
19	<p>1. Ukuran benih kurang dari 2 cm (1)</p>	Ikan Mati	Ikan Terserang Parasit, Ikan Mati	0
20	<p>1. Kolam fiber (1)</p> <p>2. Air tercemar H<sub>2</sub>S (0,4)</p> <p>3. Pemberian antiseptik lebih dari 2kali/minggu (1)</p> <p>4. Pembuangan kotoran 2-3kali/minggu (1)</p> <p>5. Ukuran benih dibawah 2cm (1)</p>	Ikan Sehat	Ikan Sehat	1

Berdasarkan Tabel 6.9 dilakukan perhitungan akurasi menggunakan Persamaan 2.11 dan menghasilkan nilai akurasi sebagai berikut.

$$\text{Nilai akurasi} = \frac{16}{20} \times 100 \% = 80\%$$

Akurasi sistem sesudah di hybrid dengan PSO berdasarkan 20 data uji adalah sebesar 90%. Nilai presentase diperoleh dari pembagian data yang benar sebesar 18 dari 20 kasus sebenarnya. Nilai akurasi tersebut diperoleh dikarenakan pada percobaan 5 dan 19 terjadi perbedaan hasil keluaran sistem dengan pakar dimana pada percobaan 5 hasil dari pakar adalah ikan terserang bakteri sedangkan keluaran sistem adalah ikan terserang parasit. Selain itu pada percobaan 19 juga terjadi perbedaan hasil keluaran sistem dengan pakar. Sedangkan sebelum di hybrid keluaran sistem menghasilkan akurasi 80% . Hal ini disebabkan karena kedua indikasi mempunyai fakta-fakta yang mirip sehingga sistem sedikit kesulitan dalam melakukan prediksi. Ketidakakurasian sistem ini juga dapat disebabkan oleh subjektifitas nilai cf pakar yang didapat dari perhitungan PSO maupun pakar itu sendiri.







## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil pengujian dan pembahasan dari implementasi algoritma *Particle Swarm Optimization-Certainty Factor* Untuk Mengenal Kondisi Ikan Lele maka diperoleh kesimpulan:

1. Algoritma *Particle Swarm Optimization – Certainty Factor* dapat diimplementasikan untuk mengenali kondisi ikan lele yang memberikan pengetahuan berdasarkan pengalaman mencoba-coba dari kombinasi fakta budidaya yang menghasilkan dampak indikasi dari percobaan. *PSO-Certainty factor* bekerja untuk memprediksi indikasi berdasarkan kombinasi fakta budidaya ikan lele. Hal tersebut dilakukan dengan mendefinisikan partikel sebagai representasi penyelesaian. Panjang partikel adalah banyaknya fakta yang dipilih sehingga jika fakta yang dipilih berjumlah 4 maka panjang inisialisasi partikel pertama merepresentasikan fakta 1 dan seterusnya sejumlah fakta yang dipilih. Setiap inisialisasi partikel, pembangkitan nilai awal diisi berdasarkan nilai bobot dari pakar sehingga jika pada bobot dari pakar bernilai 0(nol) yang artinya tidak ada pengaruh terhadap indikasi maka tidak akan dibangkitkan nilainya. Selama proses optimasi, kecepatan dan posisi partikel selalu diperbarui. Solusi penyelesaian berupa nilai cf pakar optimum yang merupakan posisi terbaik yang pernah dicapai sebuah partikel sampai dengan iterasi tertentu. Posisi terbaik yang pernah dicapai tersebut yang dijadikan sebagai nilai cf pakar pada algoritma *Certainty Factor* untuk diproses lebih lanjut sampai menghasilkan kesimpulan sistem berupa indikasi.
2. Parameter PSO yang telah diuji sebelumnya di masukan ke sistem guna pengujian akurasi sistem. Berdasarkan perbandingan pada pengujian akurasi, Algoritma *PSO-Certainty Factor* terbukti melakukan prediksi lebih baik dengan akurasi 90%. Hal itu membuat algoritma *PSO-Certainty Factor* mengalami peningkatan akurasi sebesar 10%.

### 7.2 Saran

Penelitian tentang Implementasi Metode *PSO-Certainty Factor* Untuk Mengenal Kondisi Ikan Lele dapat dikembangkan dengan beberapa saran yaitu

1. Pada saat menentukan kecepatan partikel yang sesuai, dapat dilakukan metode perhitungan lebih lanjut agar interval kecepatan sesuai dengan karakteristik permasalahan sehingga PSO dapat menghasilkan solusi yang lebih baik.
2. Untuk tampilan sistem bisa dikembangkan lebih lanjut dengan penambahan animasi agar lebih menarik dan mudah dipahami bagi pengguna.

## DAFTAR PUSTAKA

- Direktorat Jenderal Perikanan Budidaya, 2013. Laporan Tahunan Direktorat Produksi Tahun 2013, Direktorat Jenderal Perikanan Budidaya, KKP. Tersedia di: <  
<https://www.djpb.kkp.go.id/public/upload/download/Pustaka/06PUSTAKA/LAPTAH%20PRODUKSI%20%202013.pdf>> [Diakses 23 September 2016]
- Direktorat Jenderal Perikanan Budidaya, 2016. Laporan Kinerja 2016 Direktorat Jenderal Perikanan Budidaya. Tersedia di: <  
[https://www.djpb.kkp.go.id/index.php/mobile/arsip/c/499/LAPORAN-KINERJA-PERIKANAN-BUDIDAYA-TAHUN-2016/?category\\_id=>](https://www.djpb.kkp.go.id/index.php/mobile/arsip/c/499/LAPORAN-KINERJA-PERIKANAN-BUDIDAYA-TAHUN-2016/?category_id=>) [Diakses 3 Januari 2017]
- Eka, M., 2016. Hybrid Particle Swarm Optimization dan K-Means untuk Clustering Data Penentuan UKT.
- Engelbrecht, A. P., 2007. *Computational Intelligent*. Second Edition penyunt. South Africa: s.n.
- Muliantara, A., 2012. Penentuan Komposisi Bahan Pakan Ikan Lele Yang Optimal Dengan Menggunakan Metode IWO-Subtractive Clustering. *Jurnal Ilmu Komputer*, 5(2), pp. 23-28.
- Permana, K. E. & Hashim, S. Z. M., 2010. Fuzzy Membership Function Generation using Particle Swarm Optimization. *International Journal Problems Compt. Math.*, 3(1).
- Rohajawati, S. & Supriyati, R., 2010. Sistem Pakar: Diagnosis Penyakit Ungas Dengan Metode Certainty Factor.
- Shi, Y., 2016. A Modified Particle Swarm Optimizer.
- Tuegeh, Maickel , Soeprijanto dan Purnomo, Mauridhi H. 2009. *Optimal Generator Shceduling Based On Particle Swarm Optimization*. *Seminar Nasional Informatika 2009 (SemnasIF 2009) ISSN: 1979-2328 Yogyakarta, 23 Mei 2009*.
- Valle, Y. D., 2008. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. Volume 12, p. 2.
- Yastita, S., Lulu, Y. D., Sari, R. P, 2012. *Sistem Pakar Penyakit Kulit Pada Manusia Menggunakan Metode Certainty Factor Berbasis Web*. *Seminar Nasional Teknologi Informasi , Komunikasi dan Industri (SNTIKI) 4. ISSN : 2085-9902 Pekanbaru, 3 Oktober 2012*.